



BACHELORARBEIT

Richard Kästner

Ingex in der Fernseh-Außenübertragung -
Dateibasierte Aufzeichnung in eine non-lineare
Schnittumgebung auf Basis von
Open-Source-Software

Ingex in outside broadcasting - File-based
recording for non-linear editing systems on the
basis of open-source-software

2011

Fakultät Medien

BACHELORARBEIT

Ingex in der Fernseh-Außenübertragung -
Dateibasierte Aufzeichnung in eine non-lineare
Schnittumgebung auf Basis von
Open-Source-Software

Autor:

Richard Kästner

Studiengang:

Medientechnik

Seminargruppe:

MT08wF-B

Erstprüfer:

Prof. Dr.-Ing. Rainer Zschockelt

Zweitprüfer:

Dipl.-Ing. Jens Hoppe (MDR)

Leipzig, 22. Juli 2011

Referat

Diese Arbeit beschäftigt sich mit der Realisierung eines mehrkanaligen Videoaufzeichnungsservers auf Basis der Open-Source-Software „Ingex“. Das Aufzeichnungssystem bietet mittels Standard-IT eine Schnittstelle von SDI-basierter Broadcast-Ausrüstung zum dateibasierten Arbeiten. Es ermöglicht die direkte Vernetzung mit einer non-linearen Schnittumgebung, wobei maßgeblich die offenen, standardisierten Austauschformate MXF und AAF genutzt werden. Zur Anpassung auf die Bedürfnisse der Fernseh-Außenübertragung wurde das System um eine Echtzeit-Schnittmöglichkeit erweitert (Edit-While-Capture). Zunächst wird in der Bachelorarbeit ein Überblick über dateibasiertes Arbeiten, Austauschformate und Metadaten gegeben. Ein besonderer Schwerpunkt liegt auf dem Material Exchange Format (MXF). Die Software Ingex wird genau analysiert, wobei der Autor auch auf die Besonderheiten frei verfügbarer Software eingeht. Folgend sind die Realisierung eines Videoservers auf Debian-Linux-Plattform und die Soft- und Hardware-Anforderungen herausgearbeitet. Abschließend wird speziell der Einsatz von Ingex in der Außenübertragung und die notwendigen Anpassungen beschrieben.

Bibliographische Angaben

Kästner, Richard:

Ingex in der Fernseh-Außenübertragung - Dateibasierte Aufzeichnung in eine non-lineare Schnittumgebung auf Basis von Open-Source-Software. - 2011. - 101 S.

Hochschule Mittweida, Fakultät Medien, Bachelorarbeit

Inhaltsverzeichnis

Referat	1
Bibliographische Angaben	1
Inhaltsverzeichnis	2
1 Einleitung	5
2 Grundlagen des dateibasierten Arbeitens	7
2.1 Merkmale dateibasierter Workflows	7
2.2 Containerformate in non-linearen Produktionssystemen . . .	12
2.2.1 Material eXchange Format	13
2.2.1.1 Eckpfeiler der MXF-Spezifikation	14
2.2.1.2 Aufbau	17
2.2.1.3 Descriptive Metadata Scheme (DMS)	20
2.2.1.4 Operational Patterns	20
2.2.1.5 OP-Atom	22
2.2.1.6 OP1a	23
2.2.1.7 Einsatzgebiete des MXF	24
2.2.1.8 Probleme im Umgang mit dem MXF	25
2.2.1.9 MXF AS02	26
2.2.2 QuickTime	27
2.2.2.1 Funktionen	28
2.2.2.2 Struktur	29
2.3 Metadaten	30
2.3.1 Strukturelle Metadaten	30
2.3.2 Deskriptive Metadaten	31
2.3.3 Formate	31
2.3.3.1 Broadcast Metadata Exchange Format . . .	31
2.3.3.2 Advanced Authoring Format	32

3	Dateibasierte Aufzeichnung mit Ingeg	35
3.1	Entstehung von Ingeg als Werkzeug zur automatisierten Studioproduktion	36
3.2	Migration von SD zu HD	38
3.3	Besonderheiten der Open-Source-Entwicklung	39
3.4	Modularer Aufbau	40
3.4.1	Capturing	40
3.4.2	FFmpeg	43
3.4.3	Writeavidmxf	43
3.4.4	Recorder	44
3.4.5	Ingeggui	46
3.4.6	Createaaf	48
3.4.7	Database und WebIngeg	49
3.4.8	Directors Cut	50
3.4.9	Quadsplit	52
3.5	Unterstützte Formate und Codecs	53
3.6	TransferToP2	56
4	Aufbau und Konfiguration eines Ingeg-Videoservers auf Debian-Plattform	57
4.1	Das Betriebssystem Debian als Plattform	58
4.2	Anforderungen an die Hardware	58
4.2.1	Mehrkernprozessor und Arbeitsspeicher	59
4.2.2	Speicher	59
4.2.3	Netzwerk	60
4.2.4	Capture-Karten	61
4.3	Eckdaten des Prototyps	62
4.4	Zusätzliche Software	63
5	Einsatz in der Außenübertragung	66
5.1	Allgemeine Grundgedanken	66
5.2	Edit-While-Capture	67
5.2.1	Analyse eines Vergleichssystems	68
5.2.1.1	Auslesen der MXF- und AAF-Container	69
5.2.1.2	Auswertung der Dumps	69
5.2.2	Zugriff auf anwachsende MXF-Dateien mit Avid	72
5.2.3	Edit-While-Capture mit AAF-Dateien realisieren	73

5.2.4	Anpassungen am Ingex-Workflow	75
5.3	Automatisierung	76
5.4	Bedienung und Konfiguration	76
5.5	Praxistests zu Belastbarkeit und Grenzen des Systems . . .	77
5.5.1	Anzahl der Aufzeichnungskanäle	78
5.5.2	Encoding über Netzwerk	80
5.5.3	Zugriff mehrerer Avid-Clients auf Material per Edit-While-Capture	82
6	Ausblick	84
7	Fazit	86
8	Selbstständigkeitserklärung	88
9	Danksagung	89
	Literaturverzeichnis	90
	Abbildungsverzeichnis	95
	Tabellenverzeichnis	97
	Anhang	98

1 Einleitung

Seit einigen Jahren ist die dateibasierte Aufzeichnung von audiovisuellen Inhalten durch sogenannte Videoservert im Fernsbereich etabliert. Häufig sind noch hohe Anschaffungskosten und proprietäre Soft-/Hardware nötig, um einen breiten Umfang professioneller Funktionen zu erreichen.

Mittlerweile existieren Open-Source-Produkte wie z.B. *Ingex* von der *British Broadcasting Corporation* (BBC), die in Kombination mit leistungsfähiger Computertechnik „von der Stange“ eine ernstzunehmende Alternative dazu darstellen. Bezüglich technischer Ansprüche bei öffentlich-rechtlichen Sendeanstalten und Fernseh-Außenübertragungen sind sie kommerziellen Produkten ebenbürtig.¹

Während meines Praktikums bei der *Media Mobil GmbH* (MMG) arbeitete ich daran, mit Hilfe der *Ingex*-Software einen Videoservert zu installieren und für den Einsatz in Fernseh-Außenübertragung nutzbar zu machen. Die MMG ist ein technischer Dienstleister für Fernseh-Außenübertragungen, der zahlreiche Live-Events wie z.B. Sport- und Politikereignisse überträgt. Meine Hauptaufgabe war die Schaffung einer neuen, herstellerunabhängigen Schnittstelle von SDI-basierter Broadcast-Technik zur dateibasierten Schnitt- und Speichertechnik. HD- und SD-Signale sollten in Echtzeit und automatisiert in Dateiform bereitgestellt werden. Der Videoservert muss zudem gängige Broadcast-Codecs unterstützen und simpel bedienbar sein.

Eine besondere Herausforderung in der Fernseh-Außenübertragung ist die dem Ereignis nahe Bearbeitung von Material mit non-linearen Schnitt-Systemen. Daher widme ich mich in meiner Bachelorarbeit der mehrkanaligen, dateibasierten Aufzeichnung in eine non-lineare Schnittumgebung. Ich habe das *Ingex*-Programm so modifiziert, dass Material bereits während der Aufnahme geschnitten werden kann (*Edit-While-Capture*). Dadurch steht einem Cutter mehr Zeit zur kreativen Bearbeitung von Inhalten zur Verfügung. Knappe Fristen können während einer Produktion leichter eingehalten werden.

¹ Gespräch mit Andreas Jasper, Schnitt-Administrator der Media Mobil GmbH, 12.07.2011

1 Einleitung

Die Arbeit ist in vier Schwerpunkte gegliedert. Zunächst werden in Kapitel 2 die Grundlagen des dateibasierten Arbeitens dargelegt. Datencontainer, insbesondere das *Material eXchange Format*, und der Umgang mit Metadaten werden beschrieben. Im dritten Kapitel erläutere ich die Software Ingex und lege ihre detaillierten Funktionsweisen dar. Die Besonderheiten von *Open-Source-Software* werden hierbei verdeutlicht. Kapitel 4 stellt den technischen Aufbau des Ingex-Videoserver vor. Aufbauend auf die beschriebene Soft- und Hardware ist im fünften Kapitel schließlich die Entwicklung eines in der Außenübertragung einsatzfähigen Prototypen beschrieben. Ich erläutere automatisierte Workflows und die Realisierung von Edit-While-Capture. Abschließend prüfe ich die Leistungsfähigkeit des Systems und lege bestehende Beschränkungen dar.

2 Grundlagen des dateibasierten Arbeitens

Allgemein ist zu beobachten, dass sich Fernsehanstalten von der band-basierenden Fernsehproduktion lösen und auf dateibasiertes Arbeiten umsteigen. „*Zukünftige TV-Produktionen werden sich eher auf vernetzte Dateiserver mit Festplattenspeichern, als auf bandbasierende Speicher stützen*“, stellte Nick Wells, Mitentwickler des mittlerweile etablierten Austauschformats MXF (*Material eXchange Format*), bereits 2003 fest.² Die Bandaufzeichnung auf Videokassetten war jahrelang Standard in der Fernseh-Industrie. Anfangs dienten analog beschriebene Bänder als Speichermedien. Mit dem Aufnahmeformat *D-1*³ etablierten sich ab 1986/87 digitale Bandformate.

Mittlerweile ersetzen dateibasierte Speicher mehr und mehr das Band als Medium. Dabei stellen sich Produzenten und Dienstleister neuen Herausforderungen. Feste Arbeitsabläufe existieren kaum, es entstehen unerwartete Kosten und Interoperabilität ist häufig nur ungenügend vorhanden.⁴

Folgend sollen die Besonderheiten dateibasierter Workflows dargelegt werden und ein Überblick über Arbeitsweisen, Technologien und Formate verschafft werden. Ich gehe auf Containerformate, besonders auf das *Material eXchange Format*, Metadaten und das *Advanced Authoring Format* (AAF) ein.

2.1 Merkmale dateibasierter Workflows

Dateibasiertes Arbeiten ermöglicht zentrale Speicher mit digitalen Inhalten von und für verschiedene Anwendungen. Ein Zugriff auf diese ist jederzeit

²Vgl. Wells: The MXF Book S. 2

³Vgl. Sony History: The Beginning of the Digital Age, unter: <http://www.sony.net/SonyInfo/CorporateInfo/History/SonyHistory/2-04.html> (abgerufen am 27.06.2011)

⁴Vgl. BBC: White Paper (WHP) 155, S. 5

möglich. Dadurch ergeben sich einige Besonderheiten.

So entfällt das Einspielen und Zuspieren von Bändern. Bei Produktionen kann direkt auf das Aufzeichnungsmedium zugegriffen und Beiträge geschnitten werden. Im Gegensatz zum Band ist ein *non-linearer* Zugriff möglich. Non-linear bedeutet, dass Material ohne größeren Aufwand an jeder beliebigen Stelle eingefügt oder entfernt werden kann.

Bänder können in der Regel nur in Echtzeit vervielfältigt werden, um den Inhalt mehrfach nutzbar zu machen. In der dateibasierten Welt ist das Kopieren von Material um ein Vielfaches schneller möglich. Dadurch ergeben sich Zeitersparnisse beim Austausch von Inhalten. Der Materialaustausch findet zudem verlustfrei statt. Dateispeicher lassen sich in der Regel einfach erweitern und sind skalierbar. Die einfache Vervielfältigung erhöht die Datensicherheit des Materials.

Nachteilig wirken sich die uneinheitlichen mobilen Datenspeicher aus. Einige Hersteller nutzen ihre eigenen proprietären Formate, wie z.B. Panasonic (*P2*) und Sony (*XDCAM*). Die Schnittstellen der Formate entsprechen Standard-IT Komponenten, jedoch kann das Trägermedium nur vom Hersteller bezogen werden oder ist lizenzrechtlichen Regularien unterlegen. Die Anschaffungskosten sind hier dadurch zum Teil höher als im Bandbereich. Erst nach konsequenter Nutzung innerhalb der dateibasierten Arbeit lassen sich durch Synergieeffekte Kosten einsparen.⁵ Zudem herrscht auch im Bereich der Codecs und Aufzeichnungsformate teilweise Inkompatibilität zwischen verschiedenen Herstellern.

Die Migration zum bandlosen Arbeiten ging einher mit der raschen Entwicklung der *Informationstechnik* (IT). Mehr Speichertiefe, höhere Rechenleistung und leistungsfähigere Netzwerke ermöglichten den Einsatz von Videovern als Ersatz für Magnetband-Aufzeichnungsmaschinen (*MAZen*). Produkte mit proprietärer Hardware sind verbreitet, es existieren aber auch bereits Aufzeichnungslösungen wie Ingeg auf Basis von Standard-IT.

Videodaten enthalten große Mengen an Informationen. Unkomprimierte High-Definition-Signale haben eine Datenrate von rund 1,5 GBit/s.⁶ Das Kodieren und Dekodieren dieser erfordert immense Rechenkapazitäten. Außerdem müssen trotz Datenreduktion große Festplattenspeicher verfügbar sein. Beispielsweise ist für eine qualitativ hochwertige Aufzeichnung einer

⁵Auskunft von Jens Hoppe, MDR, 20.07.2011

⁶gemäß SMPTE 292M

2 Grundlagen des dateibasierten Arbeitens

Stunde Materials im Codec *DNxHD 185*⁷ folgender Speicher nötig:

$$\frac{185 \text{ MBit/s} * 60 * 60}{8 * 1000} = 83,25 \text{ GB/h}$$

Schon einige Stunden DNxHD-Material überschreiten die Kapazitäten handelsüblicher Datenträger. Daran wird deutlich, dass dateibasierte Aufzeichnung noch vor einigen Jahren an die Grenzen der damals verfügbaren Technik stieß, die über deutlich geringere Speichertiefen verfügte. Um mit den enormen Datenmengen umgehen zu können, hat sich im Bereich der non-linearen Fernsehproduktion ein *hierarchisches Speichermanagement* etabliert. Die einzelnen Speicherebenen lassen sich kategorisieren in *Online*, *Nearline*, *Archiv* und *Offline*.

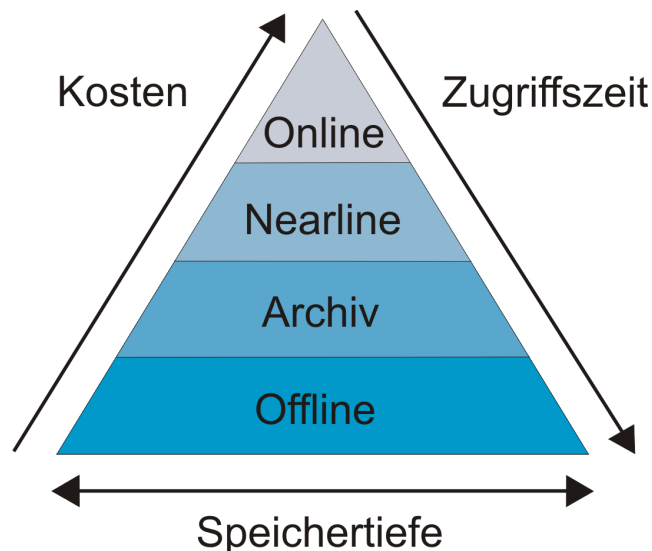


Abbildung 2.1: Der hierarchische Aufbau eines Speichersystems (Quelle: eigene Darstellung)

Eine Speicherebene steht umso höher, je schneller die gespeicherten Daten benötigt werden. Zugriffszeiten für Onlinesysteme sind auf Geschwindigkeit maximiert, während Offline-Material in der Regel erst nach einiger Zeit verfügbar sein muss. Üblicherweise steigt die Gesamt-Speichergröße mit niedrigerer Speicherebene, während die Kosten pro Megabyte abnehmen.

⁷DNxHD ist der HD-Codec des verbreiteten Schnittprogramms AVID

Online Gemeinsam genutzter Speicher ist ein Schlüsselmerkmal der Architektur vernetzter Produktionsumgebungen.⁸ Im hierarchischen Aufbau stehen zuoberst die Online-Speicher. Das sind teure Festplattenspeichersysteme mit hohem Datendurchsatz und geringer Latenz. Sie sind unmittelbar mit produktiven Geräten verbunden und für die direkte Materialbearbeitung geeignet.⁹

Online-Speicher bilden eine Brücke zu linearen Prozessen wie einem direkten Leitungseinspiel über eine serielle Video-Schnittstelle. Sie garantieren die Verfügbarkeit von Inhalten auf Abruf. Dadurch kommen sie u.a. in den Bereichen Sendeabwicklung (SAW), Produktion, Postproduktion und Ingest zum Einsatz. In der SAW werden fertige Beiträge der Sendeautomation zur Verfügung gestellt. Die Postproduktion hingegen nutzt Material online an den Videoschnittplätzen. Über einen geeigneten Ingest ergibt sich die Möglichkeit, bereits während der Aufnahme auf anwachsendes Material zugreifen zu können (*Edit-While-Capture*, siehe Punkt 5.2 auf S. 67).

Nearline Als Erweiterung lassen sich Nearline-Systeme nutzen, die kürzlich genutztes Material sichern. Das sind zusätzliche Datenspeicher, von denen sich unmittelbar Material in die Online-Systeme überführen lässt. Hier werden kostengünstigere Festplattenspeichersysteme eingesetzt¹⁰, in der Regel RAID-Systeme. Nearline-Speicher werden zudem genutzt werden, um Archivmaterial zwischenzuspeichern.

Archiv Wenn die Kapazitäten der Nearline-Systeme erschöpft sind, muss Material zur langfristigen Aufbewahrung gelagert werden. Dazu nutzt man typischerweise Bandrobotersysteme und Magnetband-Medien wie z.B. *LTO*, um automatisiert auf eine große Menge an Inhalten zugreifen zu können. LTO steht für Linear Tape Open und bezeichnet einen verbreiteten Typ von Magnetbändern, der zur Datensicherung und Archivierung eingesetzt wird. Ebenso sind aber auch optische Datenträger im Einsatz.¹¹

Bandbasierte Archivmedien haben den Vorteil, dass sie für geringe Kosten große Speichermengen und hohe Zugriffsgeschwindigkeiten bieten. Vor

⁸Vgl. Walsh/Geppert: Storage Management, FKT 12/2006 S. 751

⁹Vgl. Oermann/Jäschke/Dittmann: Vertrauenswürdige und abgesicherte Langzeitarchivierung multimedialer Inhalte S. 28

¹⁰Vgl. Walsh/Geppert S. 750

¹¹Ebd. S. 750

der Nutzung von Archivmaterial erfolgt ein Kopiervorgang in ein Online- oder Nearline-System.

Das Speichern von Metadaten ist essenziell in Archiven, um Material später finden und zuordnen zu können. Metadaten werden genauer unter Punkt 2.3 auf S. 30 erläutert.

Offline Optische Medien oder ausgelagerte Bänder, die im Archiv keinen Platz mehr haben, fallen unter die Kategorie Offline. Auf sie ist kein automatisierter Zugriff mehr möglich.

Der Preis und die Einfachheit der Handhabung der einzelnen Systeme nimmt von oben nach unten über die vier Speicherebenen ab (siehe Abbildung 2.1 auf S. 9). Umgekehrt proportional dazu verhält sich die Kapazität und Latenz in den Speicherebenen.¹² Die einzelnen Speicher müssen so gewichtet werden, dass bei minimalen Kosten jede Ebene die nötige Kapazität zur Verfügung hat. Diesen Prozess wird als Systemauslegung bezeichnet.¹³

Der Rundfunk stellt hier spezielle Anforderungen. Es müssen klare Regeln herrschen, wann, wie und von wem Material zwischen den Speicherschichten bewegt wird. So muss ein Spielfilm nur zu bestimmten Sendeterminen in den Online-Speicher überführt werden, während andere Inhalte, wie z.B. bestimmte Trailer oder Sendehinweise dauerhaft zur Verfügung stehen müssen.

	Medium	Zugriffszeit	Kosten pro Byte
Online	Festplatten-RAID-Systeme, SCSI, Fibre Channel	Sofort	Hoch
Nearline	Festplatten-RAID-Systeme, SATA	Sekunden	Mittel
Archiv	Automatisierte Band-Bibliothek, optische Jukebox	Minuten bis Sekunden	Niedriger
Offline	Wechseldatenträger wie alte Bänder, CDs oder DVDs, die in gesondertem Lager aufbewahrt werden	Stunden bis Tage	Gering

Tabelle 2.1: Eigenschaften der einzelnen Speichersysteme (Quelle: Austerberry S. 265)

Einen wichtigen Aspekt stellt dementsprechend auch die Priorisierung

¹²Vgl. Oermann/Jäschke/Dittmann S. 28

¹³Vgl. Walsh/Geppert S. 750

des verwalteten Materials dar. Sendematerial muss beispielsweise schnellstmöglich auf dem Sendeserver bereitgestellt werden. Daher steht der *Play-out*¹⁴ an oberster Stelle. Wer Material anfordert, muss gleichzeitig die Gewissheit haben, wann dieses verfügbar ist. Daher ist für den Benutzer ein transparentes Speichermanagement von Nöten.¹⁵

Abschließend lässt sich sagen, dass Content durch einen vierstufigen Speicherprozess wesentlich schneller als bisher zur Verfügung steht. Über ein Netzwerk kann nun durch die verbesserten Datenraten audiovisuelles Material standortunabhängig und in hoher Qualität abgerufen werden.

2.2 Containerformate in non-linearen Produktionssystemen

Die Basis dateibasierter Workflows sind multimediale Datenformate, sogenannte Container oder Wrapper. Sie beinhalten audiovisuelle und beschreibende Informationen. Container fügen verschiedene Datentypen korrekt zusammen ohne diese durch Transkodieren verändern zu müssen.

Im Bereich professioneller Videotechnik finden sich zahlreiche verschiedene Container. Sowohl proprietäre Formate wie das Apple *QuickTime*, welches gleichermaßen im Heimanwenderbereich und professionell genutzt wird, als auch offene Formate wie das *Material eXchange Format*, das speziell für den professionellen Anwendungsbereich entwickelt wurde.

Für die Fernsehproduktion sind bestimmte Eigenschaften eines Containers von Bedeutung. Der reibungslose Austausch von Inhalten zwischen verschiedenen Plattformen soll gewährleistet sein. Dazu werden robuste, flexible Formate benötigt, die von internationalen Organisationen standardisiert und zertifiziert sind. Das sichert eine beständige Weiterentwicklung der Container. Zusätzlich benötigt man spezielle Funktionen wie Streaming-Fähigkeit, das Referenzieren auf externe Quellen und umfassende Metadaten (siehe Punkt 2.3 auf S. 30)

Im Folgenden gehe ich vor allem auf zwei in der Fernsehwelt verbreitete Formate ein, MXF und QuickTime.

¹⁴Playout ist die englische Bezeichnung für das Zuspielen von Sendematerial

¹⁵Vgl. Walsh/Geppert S. 750

2.2.1 Material eXchange Format

Mit dem Einzug des dateibasierten Arbeitens stieg auch die Nachfrage nach einem unabhängigen, standardisierten Format für audiovisuelle Inhalte und Metadaten. Herstellerbezogene Container wie Windows-AVI (*Audio Video Interleave*) und das Apple *QuickTime* erfüllten bestimmte Anforderungen, waren jedoch auf viele professionelle Anwendungen nicht zugeschnitten. Speziell die Mitführung von ausführlichen Metadaten im Programmmaterial ist von hoher Bedeutung.¹⁶

Die Konvergenz von Fernsehtechnik, Computer- und Telekommunikationstechnologie erforderte eine format-, kompressions- und plattformunabhängige Verbreitungsmöglichkeit von AV-Inhalten mittels IT-Systemen.¹⁷ Für effiziente, kostengünstige TV-Produktion ist dazu ein Format nötig, welches weitgehend als Standard für den Datei-Austausch akzeptiert wird.¹⁸

Die *Society of Motion Picture and Television Engineers* (SMPTE) standardisierte dazu im Jahr 2003 das Material eXchange Format, kurz MXF, unter der Norm SMPTE 377M. Dem voraus ging die kontinuierliche Entwicklungsarbeit von Experten aus aller Welt. Bereits 1996 hatte sich das *Professional MPEG-Forum* mit zahlreichen Herstellern und später auch der *AAF Association* zusammengesetzt, um gemeinschaftlich ein offenes Austauschformat nach den Bedürfnissen der professionellen Fernsehindustrie zu schaffen. Unterstützung erhielten sie von der *European Broadcasting Union* (EBU) und der SMPTE.

Sie formulierten damals folgende Ziele:¹⁹

1. Der Transport programmbezogener Metadaten und der Audio-/Video-Komponenten.
2. Dateien sollen bereits zu bearbeiten sein, wenn sie noch nicht fertig transferiert sind.
3. Nützliche Informationen sollen auch dekodierbar sein, wenn Teile einer Datei fehlen, z.B. durch eine unvollständige Übertragung.
4. Das Format soll offen, standardisiert und kompressionsunabhängig sein.

¹⁶Vgl. Wells: The MXF Book S. 2

¹⁷Vgl. Schmidt: Datenformate im Medienbereich S. 163

¹⁸Vgl. Wells S. 2

¹⁹Ebd. S. 2

5. Es soll auf den Austausch von kompletten Sendungen oder Programmteilen abzielen.
6. Über allem soll es simpel genug sein, um Echtzeit-Implementationen zu erlauben.

Die AAF Association und AVID²⁰ legten Wert darauf, dass das MXF überdies interoperabel mit dem Postproduktionsformat AAF (siehe S. 32) sei. Der Gedanke war, Metadaten im kompletten Workflow von der Postproduktion zur Distribution und dem fertigen Programm offen transportieren zu können, auch zwischen beiden Formaten. Pro-MPEG und AAF Association einigten sich auf eine *Zero Divergence Doctrine* (ZDD), um die Interoperabilität zu gewährleisten. Alle Veränderungen am Datenmodell des einen sollten auf das andere übertragen werden.

Heute spricht man daher vom MXF als Unterformat des AAF, da die schon vorhandenen objekt-orientierten Strukturen des AAF aus Kompatibilitätsgründen verwendet wurden. Das MXF beschränkt sich auf wesentliche Funktionen, um Format und Struktur so einfach wie möglich zu halten. Ein großer Umfang an komplexen Funktionen für die Postproduktion wie beim AAF ist nicht nötig und stände dem ursprünglichen Ziel entgegen: Eine möglichst simple Struktur bei Erfüllung der Anforderungen an ein leistungsfähiges Austauschformat.²¹

2.2.1.1 Eckpfeiler der MXF-Spezifikation

Die Grundlage für das MXF in seiner heutigen Form waren vor allem drei Entwicklungen: *UMID*, *KLV* und *Content Package*.

UMID Die SMPTE beschrieb schon 1998 in ihrem Abschlussbericht zu einem zukünftigen, standardisierten TV-Dateiformat einen *Unique Material Identifier* (UMID) - eine eindeutige Kennzeichnung von multimedialen Dateien und Datenströmen. Der UMID ist ein Grundpfeiler der MXF-Spezifikation.²² Inhalte lassen sich mit diesem einzigartigen Code zweifelsfrei identifizieren und referenzieren.

²⁰AVID ist ein verbreiteter Hersteller von non-linearen Schnittsystemen

²¹Vgl. Wells S. 3

²²Ebd. S. 6

Basic UMIDs sind 32-Byte-Nummern und identifizieren beispielsweise eine bestimmte Fernsehsendung. Sie enthalten fest vorgeschriebene Angaben über das audiovisuelle Material und eine eindeutige Kennzeichnung. Zusätzlich existieren noch 64 Byte große *Extended UMIDs*, die das Hinzufügen von weiteren 32 Byte an Metadaten erlauben.²³ Dabei kann es sich um beschreibende Informationen wie den Erstellungszeitpunkt, geographische Koordinaten des Aufnahmeortes und Angaben zu Land, Organisation und Personen handeln.²⁴

Der UMID dient im Allgemeinen der Identifizierung von Paketen mit audiovisuellem Inhalt. Es existieren im MXF noch zahlreiche weitere solcher Kennzeichnungen (*Identifier*): *UUIDs*, *SMPTE Universal Labels*, *local tags* und diverse Andere wie *Track IDs*.

Das MXF nutzt einzigartige globale IDs wie die *UUID* (*Universally Unique Identifier*) während der Erstellung neuer Dateien. *Track IDs* und *Track Numbers* werden ebenfalls vergeben. Diese gelten nur für eine Gruppe von Header-Metadaten und können somit in einer Datei mehrmals auftreten.²⁵

Die *Universal Labels* als registrierte, statische Werte sind zusammen mit den *local tags* unter dem folgenden KLV-Protokoll beschrieben.

Key-Length-Value-Protokoll *Key-Length-Value* (KLV) steht für Schlüssel-Länge-Wert und beschreibt ein Protokoll zur Transformation von Daten in einen seriellen Bytestrom. Das Verfahren wurde im Standard SMPTE 336M als *Data Encoding Protocol* definiert.²⁶ Es ist der Eckpfeiler der MXF-Codierung. Im Gegensatz zu sprachabhängiger Syntax-Kodierung wie z.B. XML basiert KLV auf numerischer Syntax.²⁷

Jeder Datenteil besteht aus einem Schlüssel, einer dazugehörigen Längenangabe sowie dem entsprechenden Wert. Alle KLV-Einheiten (auch *Triples* genannt) sind in einer bestimmten systematischen Ordnung gelagert.²⁸

Das Feld *Key* (Schlüssel) weist jedem Dateipaket einen einzigartigen 16 Byte großen, registrierten SMPTE-Code zu. Die ersten 4 Bytes sind dabei gleich und identifizieren den Schlüssel als *SMPTE Universal Label* (UL). Die

²³Vgl. SMPTE 330M-2004

²⁴Vgl. Liebl/Wallaszkovits: UMID S. 1

²⁵Vgl. Wells S. 21

²⁶Vgl. Schmidt S. 163

²⁷Vgl. Wells S. 16

²⁸definiert in SMPTE 377M-2004

restlichen 12 Bytes sind individuell von der SMPTE festgelegt. Der Schlüssel kann beispielsweise die in Byte 5 festgelegten Beschreibungen haben:²⁹

- 01 = verzeichnete SMPTE-Einträge (Dictionary Items)
- 02 = SMPTE-Gruppen (als Sätze oder Zusammensetzung von Metadaten)
- 03 = SMPTE-Container und -Wrapper
- 04 = Bezeichnungen (Labels)

Das nächste Feld legt die Länge (*Length*) der folgenden Daten fest. Die Notation erfolgt in *Big-Endian*. Das bedeutet, dass das Byte mit den signifikantesten Stellen zuerst gespeichert wird. Beispielsweise würde ein Timecode nach Big-Endian demzufolge von links nach rechts im Speicher adressiert sein.

Falls ein Interpreter ein Paket nicht entschlüsseln kann (sogenannte Dark Data) oder es nicht benötigt, können die Bytes mittels der bekannten Länge komplett übersprungen werden.³⁰ Dadurch ist die Funktionalität vom MXF prinzipiell erweiterbar, ohne die Leistungsfähigkeit einzelner MXF-Systeme einzuschränken. Rückwärtskompatibilität ist gegeben.³¹

Value ist schließlich der eigentlich Wert bzw. Inhalt, eine Sequenz von Bytes. Die Länge der Sequenz bestimmt das zuvor beschriebene Feld *Length*. Falls die Länge eines Values nicht bekannt ist, können bestimmte Standardwerte eingetragen werden.³² Das ist z.B. bei einem einlaufenden MPEG-Datenstrom der Fall. Dort signalisiert ein unbestimmter *Length*-Wert, dass der Bitstrom noch nicht abgeschlossen ist.

Um nicht jedem einzelnen Element einen 16 Byte großen Schlüssel zuzuweisen, lassen sich die Metadaten sinnvollerweise zu Gruppen zusammenfügen. Die vier relevanten Standards sind:³³

- **Universal sets** (alle Datensätze haben 16 Byte ULs)
- **Local sets** (häufigste Gruppe, jeder Datensatz hat kurzes lokales Schlagwort von 2 Bytes)

²⁹Vgl. Gläss/Flügel/Röhr/Rygel: Material eXchange vs. Vergleichbare Formate S. 8

³⁰Vgl. Schmidt S. 164

³¹Vgl. Nowak/Röder: Möglichkeiten des MXF-Formates bei der parallelen Produktion für verschiedene Produktionskanäle mit Virtual Set Systemen S. 81

³²Vgl. Wells S. 17

³³Ebd. S. 18

- **Variable-length packs** (Datensätze standardisiert, keine Schlagwörter, nur Länge und Wert zu setzen)
- **Fixed-length packs** (Länge und Datensätze standardisiert, keine Schlagwörter, nur Wert zu setzen)

Die Schlagwörter (*local tags*) sind festgelegte Bezeichnungen für Daten, die innerhalb einer Gruppe von Metadaten als ID dienen.

Content Package Der dritte Eckpfeiler des MXF ist das *Content Package* (CP). Es wurde ebenfalls im Zuge der Arbeit an einem standardisierten TV-Austauschformat Ende der 1990er entwickelt. Ursprünglich war es für den Datentransfer via SDTI (*Serial Data Transport Interface*)³⁴gedacht. Mit *SDTI-CP* können über die koaxiale SDI-Schnittstelle komprimierte Videosignale in synchroner Form geschickt werden. Dadurch lassen sich beispielsweise MPEG-2 datenreduzierte Videosignale überspielen, ohne sie zwischendurch dekodieren und erneut datenreduzieren zu müssen.³⁵

2.2.1.2 Aufbau

Eine MXF-Datei hat grundsätzlich eine dreigliedrige Basisstruktur aus Header, Body und Footer (Abbildung 2.2).³⁶ Die einzelnen Abschnitte (*partitions*) sollen schnellen Zugriff auf die Schlüsselteile der Datei bieten und vereinfachen das teilweise Wiederherstellen von Daten. Die wichtigsten Module im MXF sind die Header-Metadaten, der Essenz-Container und die Index-Tabelle.



Abbildung 2.2: Basisstruktur einer beispielhaften MXF-Datei (Quelle: Wells S. 24)

³⁴SDTI ist eine serielle Schnittstelle, um datenreduzierte SDI-Signale ohne (De-)Kodierung zwischen Maschinen auszutauschen

³⁵Vgl. BET Fachwörterbuch: SDTI-CP, unter: <http://www.bet.de/Lexikon/Begriffe/sdticp.htm> (abgerufen am 16.06.2011)

³⁶Vgl. SMPTE: MXF Engineering Guideline EG41 S. 4

Header Der Header umfasst die notwendigen strukturellen und optionalen deskriptiven Metadaten (ausführlich unter Punkt 2.3 auf S. 30 erklärt). Damit sind Aufbau und Inhalt der Datei sofort ersichtlich. Die strukturellen Metadaten beschreiben die Synchronisation und das Timing der Essenz (Video, Audio, Daten), die in Form von *Picture*-, *Sound*- und *Timecode*-Tracks vorliegt. *Tracks* repräsentieren dabei die zeitliche Komponente im MXF. Sie haben Eigenschaften wie *Duration* (Dauer) und *Edit Units* (üblicherweise die Framerate). Metadaten können zusätzlich auch direkt in der Essenz gespeichert werden. Beispielsweise lassen sich deskriptive Metadaten als Tracks darstellen.

Gleichzeitig legt der Header die *Packages* (Pakete) fest. Ein Package gruppiert die einzelnen Tracks, Komponenten und Timelines, um eine synchrone Darstellung zu erzeugen. Die Pakete sind in der Regel verschachtelt. Die Verschachtelung kann vom Ursprungsmaterial bis zur fertigen Komposition gehen. Das hochrangigste Paket repräsentiert die fertige Sequenz (*Material Package*), nicht die Essenz selbst (*File Package*).³⁷ Abbildung 2.3 verdeutlicht, wie das übergeordnete Material Package festlegt, welcher Teil des originalen File Packages in der Sequenz genutzt wird. In dem Beispiel wird der technische Vorspann und das Schwarzbild am Ende weggelassen.

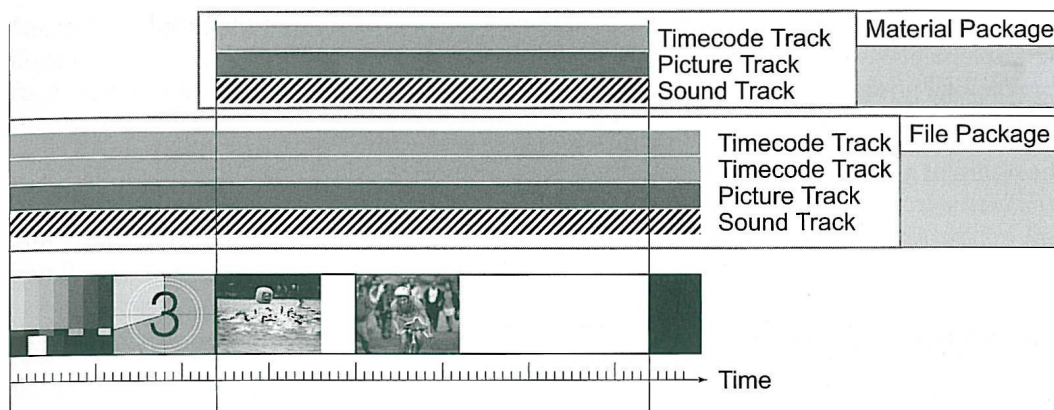


Abbildung 2.3: Das Material Package (oben) mit den abzuspielenden Tracks und das File Package (unten) mit der Originalessenz im Vergleich (Quelle: Wells S. 27)

Essenz Den gespeicherten Content eines MXF beschreibt das *top-level File Package*. Es beinhaltet die verwendete Essenz, also das Grundmate-

³⁷Vgl. Wells S. 13

rial und ist Teil der *Body-Partition*. Ältere Versionen der Essenz können als *lower-level File Package* vorhanden sein. Deren Nutzen liegt jedoch lediglich in einer Metadaten-Historie.

Essenz-Daten liegen allgemein als dem SDI-Signal ähnlicher Essenz-Container oder als Multiplex vor (ähnlich dem MPEG-Datenstrom).³⁸ Sie werden mittels KLV-Codierung (siehe S. 15) im *Generic Container* gespeichert. Das Schreiben der Daten erfolgt durch dieses Verfahren häufig frameweise und sehr schnell, was eine Implementation zum Streaming vereinfacht.³⁹ Jeder Frame wird dabei in einem eigenen *Content Package* abgelegt. Neben dem *Frame-Wrapping* existiert außerdem das *Clip-Wrapping*, bei welchem dem CP ein Clip mit der Dauer der gesamten MXF-Sequenz zugewiesen wird (Abbildung 2.4).

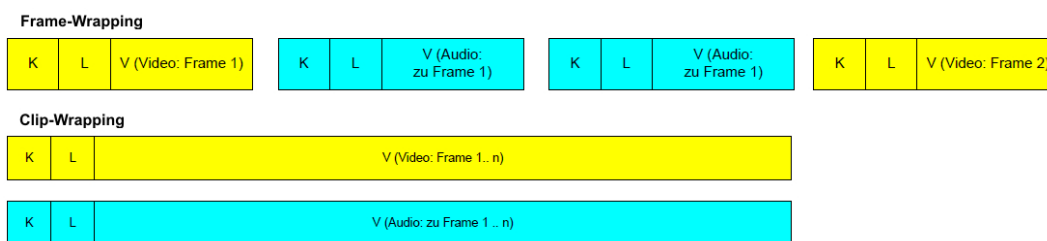


Abbildung 2.4: Frame-Wrapping (oben) und Clip-Wrapping (unten) mit ihrer unterschiedlichen KLV-Struktur (Quelle: Nufer S. 10)

Index-Tabelle Die Index-Tabellen (*index tables*) haben die Aufgabe, eine neue zeitliche Position eines Tracks im Grundmaterial (File Package) in einen Byte-Versatz umzuwandeln. Damit ermöglichen die Tabellen den schnellen Zugriff auf Material an einer zufälligen Stelle. Diese Umwandlung muss exakt sein, damit die Bild- und Tonessenz synchron wiedergegeben wird. Der Aufbau der Index-Tabelle ist abhängig vom Kompressionsformat, da sowohl variable Framegrößen (z.B. Long GoP MPEG) vorkommen können, wo jeder Frame einzeln verzeichnet sein muss, als auch konstante Framegrößen (z.B. DV) mit simpel fortlaufendem Index.⁴⁰

Footer Der Datei-Footer schließt das MXF-File ab. Er indiziert, dass eine Datei erfolgreich geschrieben wurde und enthält Informationen, die mögli-

³⁸Vgl. Wells S. 13

³⁹Ebd. S. 35

⁴⁰Vgl. Ferreira: MXF Report S. 3

cherweise noch nicht bekannt waren, als der Header erstellt wurde.

2.2.1.3 Descriptive Metadata Scheme (DMS)

Um deskriptive (beschreibende) Metadaten auszutauschen, sind standardisierte Strukturen nötig. Dazu schuf die SMPTE die Norm 380M, das *Descriptive Metadata Scheme 1*. Dort sind drei wesentliche Kategorien beschrieben:

- *Production Framework* (Beschreibungen einer gesamten Produktion, z.B. Name des Produzenten)
- *Scene Framework* (Beschreibungen einer Szene bzw. eines Inhalts, z.B. eines besonderen Ereignisses)
- *Clip Framework* (Beschreibungen des Settings bzw. der Drehbedingungen)

Die deskriptiven Metadaten liegen im MXF ebenfalls als Tracks vor. Sie charakterisieren häufig andere Tracks - Bild, Ton und/oder Timecode.

Metadata Dictionary Um Metadaten korrekt aus den KLV-codierten Daten übersetzen zu können, ist eine Art Wörterbuch vorhanden, in welchem die entsprechenden Schlüssel verzeichnet und registriert sind - das *Metadata Dictionary*. Alle SMPTE-Definitionen sind in diesem Verzeichnis offen eingetragen.⁴¹ Hersteller können sich daran orientieren, wenn sie ihre eigenen Metadaten implementieren. Mit dem RP 224⁴² entwickelte die SMPTE eine zusätzliche Festlegung, um die Universal Labels im MXF als Werte interpretierbar zu machen.

2.2.1.4 Operational Patterns

Die Anforderungen an ein audiovisuelles Austauschformat sind unterschiedlich von Fall zu Fall. Daher definiert das MXF einzelne *Operational Patterns* (OP), die verschiedene Beschränkungen an die Komplexität festlegen. Das simpelste nennt sich OP1a und transportiert ein einziges Programmsegment in einer Datei. Komplexere OPs bestehen z.B. aus mehreren Segmenten, die durch Schnitte getrennt sind. Möglich ist auch ein Programm

⁴¹Vgl. SMPTE RP 210, Metadata Dictionary Registry of Metadata Element Descriptions

⁴²Vgl. SMPTE RP 224, SMPTE Labels Registry

mit verschiedenen Untertiteln für mehrere Länder oder verschiedene Qualitätsstufen in einem OP.

Insgesamt existieren 10 unterschiedliche, spezifizierte Operational Patterns. Davon sind 9 Teil der MXF-Spezifikation in SMPTE 377M.⁴³ Sie definieren strukturelle Metadaten und Beschränkungen im MXF-Format, um Interoperabilität zwischen gleichartig komplexen Anwendungen zu sichern.⁴⁴ Im Wesentlichen beschreiben die OPs das Verhältnis zwischen Material Package (fertige Sequenz) und File Package (Materialquelle).

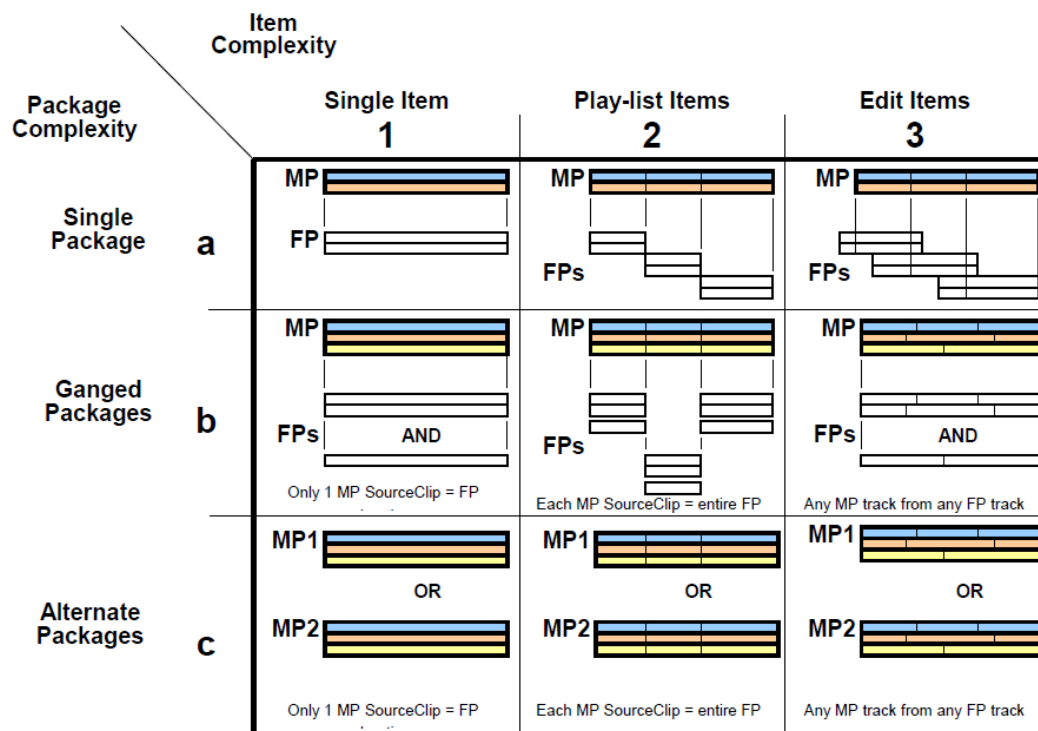


Abbildung 2.5: Die Generalized Operational Patterns mit ihrer unterschiedlichen Paket- und Abspielkomplexität im Überblick (Quelle: SMPTE Engineering Guideline S. 38)

In Abbildung 2.5 sind die 9 *Generalized Operational Patterns* aufgeführt. Sie unterscheiden sich in ihrer Funktionalität. In der Grafik steigt nach rechts hin die Komplexität in der Zeitachse des MXF. OP1-Files spielen eine gesamte Timeline aus, die mit dem top-level File Package identisch ist (*Single Items*). Daher sind sie mit einem Band vergleichbar. In der zweiten Spalte beschreibt das Material Package eine sequentielle Wiedergabeliste einzelner File Packages, die im gesamten abgespielt werden (*Play-list Items*). Im

⁴³Vgl. Wells S. 96

⁴⁴Ebd. S. 22

OP3 werden jeweils nur Teile der File Packages nacheinander wiedergegeben. Die Materialquellen sind auf diese Weise neu geschnitten. Diese sogenannten *Edit Items* arbeiten mit zufälligem Zugriff, daher sind sie in der Regel nicht streamingfähig.⁴⁵

Mit abfallender Spaltenzahl in der Grafik steigt die Komplexität der Paketverknüpfungen. Im OPa ist jedem Material Package nur ein File Package zugewiesen, im OPb hingegen lassen sich mehrere miteinander verknüpfen (*Ganged Packages*) und anhand des Material Packages synchronisieren.⁴⁶ So lässt sich beispielsweise eine externe Audiodatei mit dem Video synchron abspielen. Mehrere Material Packages, also Ausspiel-Sequenzen, realisiert das OPc. Diese *Alternate Packages* machen z.B. verschiedene Sprachversionen eines Films möglich. Ebenso lassen sich für Archivzwecke in einer Datei vollaufgelöste Versionen zusammen mit Proxies zur Voransicht speichern.⁴⁷

Um maximale Interoperabilität zu gewährleisten, müssen Operational Patterns abwärtskompatibel sein. Dadurch soll sicher gestellt werden, dass Geräte jeweils auch weniger komplexe OPs enkodieren und dekodieren.

In dateibasierten Systemen haben sich vor allem die Operational Patterns OP1a und OP-Atom etabliert. Sie bieten die simpelsten MXF-Strukturen und waren deshalb am besten geeignet, zuerst den Markt zu durchdringen.⁴⁸ Folgend werden kurz ihre Eigenschaften erläutert.

2.2.1.5 OP-Atom

Neben den 9 Generalized Operational Patterns existieren *Specialized Operational Patterns*. Diese sind auf besondere Klassen von Anwendungen zugeschnitten. Als Essenz-Container für non-lineare Schnittsysteme wurde das Operational Pattern *Atom* entworfen. Es ist in SMPTE 390M spezifiziert.

OP-Atom-Dateien besitzen nur eine Essenz. Sie referenzieren nicht auf externe Inhalte. Jeder Teil der Essenz (Video, Audio, Daten) ist in einem separaten Track und einer separaten Datei gespeichert. Dadurch lassen sich vor allem Bild und Ton leichter und schneller getrennt bearbeiten. Die Atom-Struktur ist daher besonders für non-lineares Arbeiten, bei dem das Quell-

⁴⁵Vgl. SMPTE Engineering Guideline S. 38

⁴⁶Vgl. Wells S. 34

⁴⁷Vgl. Santos: Operational Patterns ... the MXF Flavors S. 6

⁴⁸Vgl. Santos S. 5

material geschnitten und gelayert wird, geeignet.⁴⁹

Ursprünglich sind Atom-MXF für Anwendungen entwickelt, die mit kompletten, fertig aufgezeichneten Dateien arbeiten.⁵⁰ Einen Weg, noch in der Aufzeichnung befindliche Atom-Files zu nutzen, zeige ich in Abschnitt 5.2 auf Seite 67 (EWC).

Auch Videokameras zeichnen MXF OP-Atom auf, so z.B. die P2-Kameras von Panasonic.⁵¹ Schnittprogramme mit Atom-Unterstützung⁵² können dadurch ohne vorheriges Wrappen oder Transcodieren auf das Material zugreifen und es bearbeiten - die Akquisition findet non-linear statt. Das P2-System nutzt im Übrigen Clip-Wrapping an Stelle von Frame-Wrapping.⁵³ Dadurch lässt sich zu jeder Zeit sampleweise Audibearbeitung durchführen.

Technisch zu beachten ist beim OP-Atom, dass eine komplette Index-Tabelle im Footer angelegt wird. Damit ist der Einsatz als Streaming-Format mit Essenz variabler Bitrate nur in Grenzen möglich.⁵⁴ Da jede Atom-Datei einen einzelnen Essenz-Track enthält, ist eine externe Synchronisation von z.B. Video- und Audio-Essenz nötig.

2.2.1.6 OP1a

Für Anwendungen, bei denen jede Datei einen fertigen Clip repräsentieren soll, eignet sich das OP1a nach SMPTE 378M. Es ist das simpelste Operational Pattern. Die Inhalte sind als komplettes Paket mit verschachtelt gespeichertem Bild und Ton vorhanden. Damit sollte Ersatz für Bänder oder AVI-Dateien geschaffen werden.⁵⁵ Die abgespielte Sequenz (Material Package) ist identisch mit dem vorhandenen Quellmaterial (File Package). Besonders im Bereich Playout findet das OP1a durch seine simple, robuste und komplette Struktur Anwendung.

Das OP1a ist nicht nur als Austauschformat verbreitet, es wird auch erfolgreich als natives Speicherformat eingesetzt. Es findet Anwendung in

⁴⁹Vgl. Avid White Paper: MXF Unwrapped S. 5

⁵⁰Vgl. Wells S. 67

⁵¹Vgl. BET Fachwörterbuch: OP Atom, unter: <http://www.bet.de/Lexikon/Begriffe/opatom.htm> (abgerufen am 27.06.2011)

⁵²z.B. Avid Media Composer und Adobe Premiere

⁵³Vgl. Abschnitt 2.2.1.2 auf S. 17

⁵⁴Vgl. SMPTE ST 0390-2011

⁵⁵Vgl. Wells S. 98

Bandspeichern, optischen und magnetischen Festspeichern⁵⁶, z.B. im weitverbreiteten XDCAM-Format von Sony. Für den Postproduktionsbereich ist es weniger geeignet, da dort das Quellmaterial geschnitten und gelayert werden muss.⁵⁷

2.2.1.7 Einsatzgebiete des MXF

Das MXF vereinfacht die Arbeit in *netzwerk- und servergestützten Systemen* wesentlich.⁵⁸ Mit dem standardisierten Containerformat können nahezu beliebige Medien- und Metadaten transportiert bzw. gespeichert werden. Es ist zum Austausch von fertigem Programmmaterial konzipiert und unterstützt neben Filetransfer auch Streaming. Dadurch ermöglicht es einen reibungslosen Übergang von den Echtzeitanforderungen der Live-Produktion zur filebasierten Nachbearbeitung.⁵⁹ Der Austausch und die Einbindung von Metadaten ist dabei unabhängig vom Video- und Audio-Kompressionsformat.

Eine dateibasierte Umgebung auf Basis des MXF bringt *Zeitersparnisse im Austausch von Content*. Übertragungen können deutlich schneller als Echtzeit stattfinden; Essenz und Metadaten sind effektiv zusammengepackt. Nach Transfers müssen keine Metadaten manuell neu hinzugefügt werden.

Im Fernseh-Live-Betrieb muss mit Material gearbeitet werden, das gerade aufgezeichnet wird. Es ist von hoher Priorität, Dateien zu dekodieren und abzuspielen, während sie noch geschrieben bzw. transferiert werden. In MXF-Dateien aufgezeichnetes Material kann bereits bearbeitet werden, während der Schreibprozess andauert. Das *Arbeiten mit unvollständigen, anwachsenden Files* ist durch die spezielle Struktur des MXF möglich. Damit wird das dateibasierte Arbeiten in Live-Produktionen vereinfacht.

Ein weiteres Einsatzgebiet sind *Archive*. Dort ist es wünschenswert, umfangreiche Metadaten direkt mit dem Material zu speichern. Die Struktur des MXF eignet sich zudem für datenbasierte Bandspeicher.⁶⁰ Die Erweiterbarkeit der MXF-Spezifikation lässt dort einen definierten Raum für Anpassungen zu.

⁵⁶Vgl. Wells S. 99

⁵⁷Vgl. MXF Unwrapped S. 5

⁵⁸Vgl. Wells S. 4

⁵⁹Vgl. Nowak/Röder S. 78

⁶⁰Vgl. Wells S. 11

Das MXF ist als *Quasi-Streaming-Format*⁶¹ die Brücke zwischen konventionell übertragenen SDI-Signalen und dem Transfer von non-linearen Daten. Es verbindet die Bereiche Ingest/Playout mit den Bereichen Schnitt/Produktion. Über SDTI kann Material beispielsweise konventionell in Echtzeit ausgetauscht werden. Ebenso kann ein normaler Nicht-Echtzeit-Transfer in Dateiform stattfinden. Das wird durch den bereits beschriebenen Generic Container möglich.

AAF und MXF bieten durch die Zero Divergence Doctrine eine hohe *Interoperabilität*. Die enge Verbindung eines Schnitt- und eines Austauschformates hat dazu geführt, dass das MXF von vielen Schnittprogrammen genutzt werden kann. Damit werden zeitaufwendige Konvertierungen vermieden. Sinnvollerweise lassen sich Metadaten zwischen beiden Formaten tauschen. Damit wird die Produktionskette von Ingest über Editing bis Playout geschlossen.

Die KLV-Codierung ermöglicht Hardware- und Software-Systemen, unbekannte oder *irrelevante Abschnitte einfach zu überspringen*. Dadurch kann das MXF leistungsfähig für spezielle Anwendungen eingesetzt werden.

Da einige OPs Video-Schnitte und Audio-Blenden unterstützen, lassen sich von einem Programm *mehrere Versionen in eine Datei integrieren*. So lassen sich beispielsweise bestimmte Szenen für die Vorführung im jeweiligen Land herauskürzen.

Über eine teilweise *Datei-Wiederherstellung* können zudem einzelne Inhalte eines MXF-Containers nutzbar gemacht werden. Material wird mittels Index-Tabellen zugeordnet. Wenn eine Datei beschädigt ist oder ein Transfer abbricht, können die Daten über die Tabellen rekonstruiert werden.⁶² Das erhöht die *Datensicherheit*. Beispielsweise kann Material aus einer mehrfach unterbrochenen Satellitenübertragung noch genutzt werden. Des Weiteren lässt sich durch das partielle Extrahieren von Daten die Zugriffsgeschwindigkeit auf MXF-Archivmaterial erhöhen.

2.2.1.8 Probleme im Umgang mit dem MXF

Das MXF sollte die Interoperabilität verbessern, hat aber durch seinen offenen Standard neue Kompatibilitätsprobleme geschaffen. Neben den verschiedenen Operational Patterns sind inkompatible Kompressionsformate

⁶¹Vgl. Wells S. 10

⁶²Ebd. S. 13

und Metadaten-Schemen vorhanden, die den Materialaustausch behindern können.⁶³ Zu viele von Herstellern spezialisierte Operational Patterns sind mittlerweile auf dem Markt. Ein MXF-Dekoder kann häufig nicht jede Struktur verarbeiten. Daher haben sich Interoperabilitäts-Tests etabliert, wo Hersteller ihre selbst erstellten MXF-Dateien zur Überprüfung verfügbar machen und gleichzeitig fremde Dateien in ihren eigenen Systemen testen.⁶⁴ Nur wenn ein Hersteller die strengen MXF-Richtlinien an eigenen Dateien erfüllt und sich beim Dekodieren fremder konform verhält, lässt sich Interoperabilität erreichen.⁶⁵

Nutzer wollen häufig, dass ihre Ausrüstung einfach wenig Einarbeitungs- und Wartungszeit erfordert und somit einfach zu nutzen ist. Simple Funktionalität ist gefordert. Gleichzeitig sind möglichst viele Features und modernste Audio- und Videokodierung erwünscht. Zukunftsfähige dateibasierte Lösungen müssen daher erweiterbar und flexibel sein. Diese Gratwanderung soll das MXF, bei allen Problemen, unterstützen. Es definiert zudem einen robusten Container für audiovisuelle Inhalte und Metadaten. Nicht lösen soll es zugehörige Probleme wie Dateizugriff, Speicherstrukturen, Netzwerk- und Verbindungsfragen sowie das externe Abrufen von Dateien.⁶⁶ Diese liegen auf der Ebene des Netzwerk- und Speichersystems.

2.2.1.9 MXF AS02

Um das MXF-Format zu optimieren, hat die *Advanced Media Workflow Association* (AMWA)⁶⁷ praxisorientierte Veränderungen daran vorgenommen. Damit will sie den Problemen entgegenwirken, die in der Bearbeitung und Verteilung von Medienbeiträgen auftreten.⁶⁸ Insbesondere das *MXF AS02* soll für verbesserte Workflows in Medieneinrichtungen sorgen.

Das AS02 ist eine Anwendungsspezifikation (AS) des MXF, die auf die Bereiche Produktion und Archivierung abzielt. Es legt Regeln fest, wie Encoder und Decoder bestimmte MXF-Typen verarbeiten, *unabhängig* vom Kompressionsformat.⁶⁹ Dadurch kann z.B. die Größe eines Bildes abgefragt werden, ohne dass der Codec bekannt ist.

⁶³Vgl. Wells S. 4

⁶⁴Vgl. Institut für Rundfunktechnik: MXF Interoperability Plug-Fest, unter: <http://mxf.irt.de/activities/2010-11-MxfPlugFest.php> (abgerufen am 28.06.2011)

⁶⁵Vgl. Wells S. 104

⁶⁶Ebd. S. 11

⁶⁷vor 2007 als AAF Association bekannt

⁶⁸Vgl. Hedtke/Schnöll: Verteilte Plattform für Produktion und Archiv S. 21

⁶⁹Vgl. Devlin: MXF AS02 „Work in Progress“ S. 2

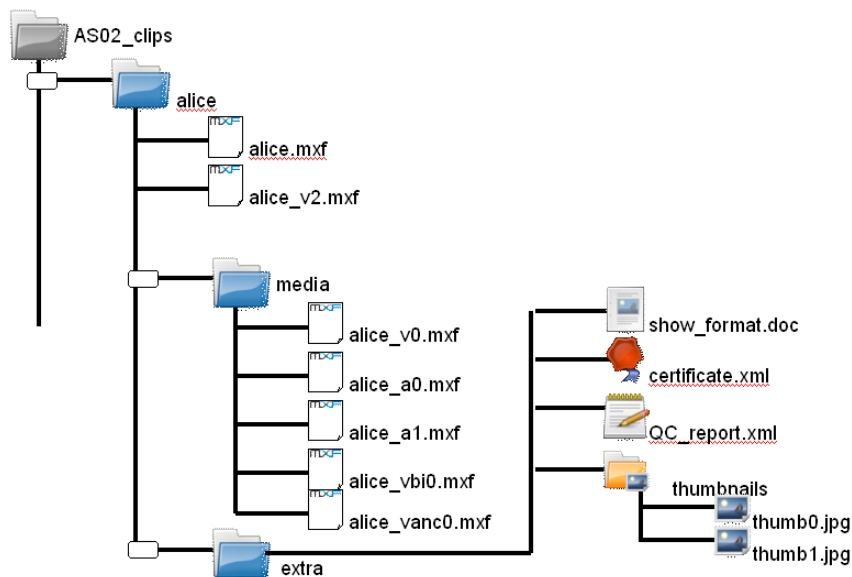


Abbildung 2.6: Die Pfadstruktur des AS02 (Quelle: Devlin S. 3)

Das AS02 soll Versionsprobleme bei der Mehrfachverwertung von Material lösen. Es legt verschiedene Varianten einer Datei wesentlich speichersparender ab. Ein sortierter Verzeichnisbaum innerhalb eines Clips sorgt dafür, dass einzelne Versionen nur über nötigste Metadaten verfügen. Sie lagern in einem extra Pfad und verweisen auf die Essenz (*media* in Abbildung 2.6). Bei der Weiterverwertung von Inhalten ist es von Vorteil, separat Informationen über die einzelnen Versionen zu speichern.⁷⁰ Dynamische Metadaten können angeben, wo sich beispielsweise Grafiken im Bild befinden bei unterschiedlichen Auflösungen oder Seitenverhältnissen.

2.2.2 QuickTime

Das Apple *QuickTime* (QT) gehört neben dem Microsoft Windows Media Format zum häufigsten Datencontainer im Heimanwenderbereich.⁷¹ Entsprechende Wiedergabe-Software ist auf jedem Mac-Computer vorinstalliert und für Windows- und Linux-Systeme verfügbar. Das QuickTime-Format hat sich ebenfalls im professionellen Bereich etabliert, beispielsweise für das Schnittprogramm *Final Cut Pro* von Apple. Üblich sind die Datei-Endungen *.mov und *.qt.

QuickTime wurde 1991 von Apple als Format für multimediale Inhalte ent-

⁷⁰Vgl. Hedtke/Schnöll S. 21

⁷¹Vgl. Schmidt S. 174

wickelt. Es war die offizielle Basis für den MPEG-4-Standard, den die *International Organization for Standardization* (ISO) festlegte.⁷² Seitdem ist das QT-Format stetig in seinem Funktionsumfang gewachsen. So können mittlerweile Referenzen auf andere verwendete Daten erstellt werden, ohne diese als neuen Datenstrom innerhalb der Datei abzuspeichern.⁷³

2.2.2.1 Funktionen

Für den Videoschnitt ergibt sich durch Referenzen die Möglichkeit, ein QuickTime anzulegen, das lediglich über Verweise auf die externen Video- und Audioströme verfügt, sowie beschreibende Daten, wie diese Ströme zusammengefügt werden. Das sogenannte *QuickTime Reference Movie* leitet im Falle eines Zugriffs auf die Referenzquellen weiter. Diese Quellen können in anderen Datenstrukturen bzw. Formaten liegen. Dadurch fällt die QT-Datei selbst sehr klein aus, obwohl sie womöglich auf mehrere Gigabyte Mediendaten verweist.⁷⁴ Eine vergleichbare Funktion hat das auf S. 32 beschriebene Advanced Authoring Format.

QT-Dateien können alle Arten zeitbasierter Daten enthalten und kombinieren.⁷⁵ Videoframes und Audiosamples ordnen sich in einzelne Spuren (*Tracks*) ein. Über *Layer* (Ebenen) können Tracks schichtweise übereinander gelegt werden. Dazu lassen sich Transparenzebenen definieren, die festlegen, welche Teile des Bildes zu sehen sind. QuickTime ist außerdem in der Lage, mehrere Tracks hintereinander als eine Sequenz abzuspielen. Überdies gibt es beschreibende Tracks, die wiederum andere modifizieren (*Modifier Tracks*) oder miteinander verbinden (*Tween Tracks*).

In den QT-Container kann ein externer Timecode geschrieben werden, der als Referenz gilt. Auch Textebenen, Grafiken, Animationen und Shockwave Flash lassen sich einbinden. Das QuickTime-Format ist zudem streamingfähig.

Im Gegensatz zum MXF lassen sich QT-Movies mit den meisten herkömmlichen Playern abspielen. Zur Identifizierung von Videokompressionsformaten nutzt QuickTime den Four Character Code (*FourCC*). Der FourCC klassifiziert eine Dateneinheit mit 4 alphanumerischen Zeichen im ASCII-Code. Daran erkennen Player, welche Codierung ein Audio- oder Videostrom

⁷²Vgl. Apple: QuickTime File Format (QTFF) S. 14

⁷³Vgl. Schmidt S. 175

⁷⁴Ebd. S. 175

⁷⁵Ebd. S. 176

hat, um ihn abzuspielen. Der Code wird ebenfalls von Microsoft verwendet, z.B. für den AVI-Container.

2.2.2.2 Struktur

Das QuickTime ist hierarchisch in einer Baumstruktur aufgebaut. Es basiert auf einem Objektmodell, dessen kleinste Einheit ein *Atom* ist. Zuoberst steht ein Root-Atom mit der ID 1. Unter ihm existieren *Parent*-Atome (Eltern), die weiter verzweigt sind in untergeordnete *Child*-Atome (Kind).⁷⁶ Atome, die nicht weiter verzweigt sind und nur Daten enthalten, haben die Bezeichnung *Leaf* (Blatt). Abbildung 2.7 veranschaulicht diese Baumstruktur.

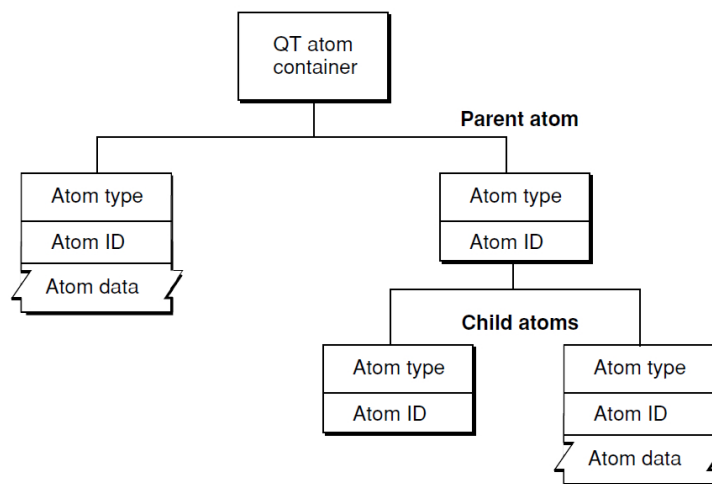


Abbildung 2.7: Eltern- und Kindatome im QuickTime-Container (Vgl. Apple: QuickTime File Format)

Die zwei Hauptverzweigungen des QuickTime sind das *Movie Data Atom* und das *Movie Atom*. Ersteres beinhaltet alle Mediendaten, also z.B. alle Frames und Samples, aber auch Verweise auf externe Dateien. Den Container für die beschreibenden Informationen und den Index der Inhalte bildet das *Movie Atom*.

Alle Atome sind Teil einer vorgegebenen Hierarchie. Die Reihenfolge der verschachtelten Atome einer Hierarchiestufe ist jedoch unwichtig, d.h. der Videoabschnitt kann auch am Anfang stehen.⁷⁷

Eine Ausnahme sind *Handler Description Atome*, die beschreiben, wie mit Datenpaketen umzugehen ist. Sie müssen ausgelesen werden, bevor Zu-

⁷⁶Vgl. QTFF S. 20

⁷⁷Vgl. Schmidt S. 177

griff auf eigentliche Daten erfolgt.⁷⁸ Daher ist es für Streaming-Anwendungen sinnvoll, Metadaten in richtiger Reihenfolge abzulegen.⁷⁹

2.3 Metadaten

In allen Stufen der Fernsehproduktion spielen Metadaten eine bedeutende Rolle. Metadaten stehen im Fernsbereich für programmbezogene Informationen, die „Daten über Daten“ liefern.⁸⁰ Das umfasst alle inhaltsbeschreibenden und technischen Informationen, die zu einer Sendung oder einzelnen Programmelementen gehören. Metadaten waren in der TV-Produktion schon immer präsent, beispielsweise durch Timecodes und Bandbegleitkarten.

Mit dem Umstieg auf dateibasierte Workflows konnte der Umfang an Einsatzmöglichkeiten deutlich erweitert werden.⁸¹ Alle notwendigen Informationen können nun leicht während der Erstellung und Nutzung von Inhalten integriert und zugänglich gemacht werden. In der Postproduktion lassen sich diese Daten später abrufen. Dabei liegen sie in elektronischer Form vor und nicht mehr in der vorher oft üblichen Papierform. Viele Firmen nutzen Content-Management-Systeme (CMS), um große Datensätze zu verwalten.⁸² Damit erleichtert sich das Auffinden und Archivieren von Material.

2.3.1 Strukturelle Metadaten

Strukturelle Metadaten sind technische Angaben, die eine Datei klammern, d.h. sie im übertragenen Sinne „zusammenhalten“. Sie kontrollieren die audiovisuelle Synchronisation und identifizieren die einzelnen Spuren und Pakete. Die Daten werden von Maschinen kreiert und sind dazu gedacht, von Maschinen gelesen zu werden. Sie stehen in einem Dateiformat in der Regel vor den deskriptiven Metadaten.

⁷⁸Vgl. QTFF S. 18

⁷⁹Vgl. Schmidt S. 177

⁸⁰Ebd. S. 150

⁸¹Vgl. Wells S. 8

⁸²Vgl. Schmidt S. 150

2.3.2 Deskriptive Metadaten

Im Gegensatz zu strukturellen werden *deskriptive* Metadaten gewöhnlich von Menschen erstellt und eingesetzt. Es handelt sich um zusätzliche beschreibende Angaben zum Material. Beim MXF-Format sind sie beispielsweise in Form von Metadaten-Tracks vorhanden, die wiederum Bild- und Ton charakterisieren. Typische deskriptive Metadaten sind Produktionsdaten wie Ort, Zeit und Projekt. Es kann sich um inhaltsbeschreibende Angaben zu einzelnen Szenen oder zu einer gesamten Sendung handeln.

2.3.3 Formate

Heute existierende Fernsehproduktionssysteme sind häufig hoch spezialisiert und in sich geschlossen.⁸³ Die übergreifende Weitergabe von Metadaten ist dadurch erschwert. Sie werden meist direkt in den Essenz-Containern transportiert. Diese unterstützen zwar, wie das MXF, umfangreiche Metadaten, dem sind aber aus Interoperabilitätsgründen bestimmte Grenzen gesetzt. Viele Hersteller erstellen daher beispielsweise *XML*-Dateien⁸⁴, um zusätzliche Daten auszutauschen und einen unkomplizierten Zugriff auf diese zu haben. Gleichzeitig sind reine Metadaten wesentlich kleiner, wodurch ein zeitsparender Austausch stattfinden kann.⁸⁵

2.3.3.1 Broadcast Metadata Exchange Format

Mit dem Ziel, den Austausch von Metadaten zu harmonisieren und zu standardisieren, entwickelte das *Institut für Rundfunktechnik* (IRT) 2003 das *Broadcast Metadata Exchange Format* (BMF). Es bietet ein umfangreiches Sortiment an beschreibendem Vokabular für die Bereiche Fernsehen, Hörfunk und Online. Zwischen IT-basierten Systemen soll das BMF die Interoperabilität steigern ohne auf spezielle Anwendungsszenarien festgelegt zu sein (siehe Abbildung 2.8). Derzeit arbeitet das IRT am BMF 2.0, das Metadaten im XML-Schema 2.0 ablegt.⁸⁶

Das BMF soll die gesamte Wertschöpfungskette der Rundfunkanstalten

⁸³Vgl. Ebner: Welche Bedeutung haben Metadaten? S. 1

⁸⁴eXtensible Markup Language (XML) ist eine standardisierte Auszeichnungssprache zur Beschreibung von Daten

⁸⁵Vgl. Schmidt S. 153

⁸⁶Vgl. IRT: Standardisierter Metadaten austausch mit BMF 2.0, unter <http://www.irt.de/de/themengebiete/produktion/bmf.html> (abgerufen am 18.06.2011)

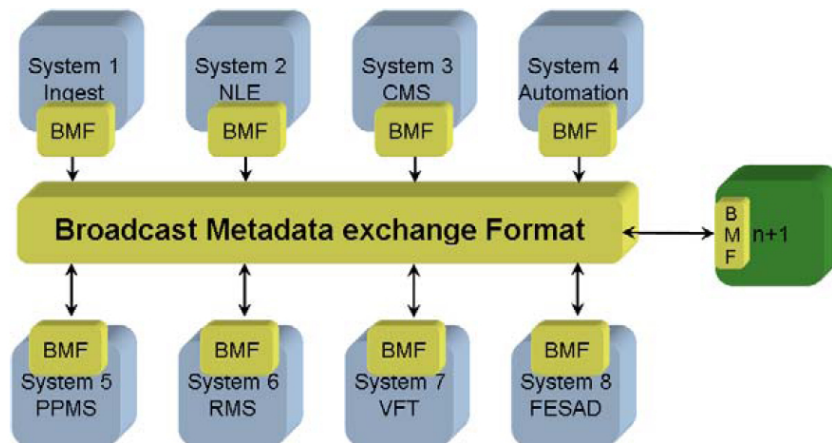


Abbildung 2.8: Im Idealfall werden alle Bereiche der Fernsehproduktion vom BMF bedient (Vgl. Ebner S. 4)

abdecken. Damit sind alle Produktionsprozesse gemeint, von der planerischen, redaktionellen Arbeit bis hin zu Akquise, Bearbeitung, Payout und Archivierung.

2.3.3.2 Advanced Authoring Format

Um umfangreiche Metadaten zwischen non-linearen Schnittsystemen auszutauschen, hat sich ein weiteres Format etabliert - das *Advanced Authoring Format* (AAF). Es wurde Ende der 1990er aus dem *Open Media Framework* (OMF) entwickelt. Man wollte im Zuge der Arbeit am MXF mit dem AAF ein Format für die Postproduktion schaffen, da für den Austausch von kreativen Projekten im Schnitt und Compositing-Bereich weitreichende Funktionen erforderlich sind. Komplexe Effekte und Bearbeitungen von Material, die das MXF aus bereits in Kapitel 2.2.1 (S. 13) genannten Gründen nicht bieten soll. AAF und MXF basieren auf dem selben Datenmodell. Sie unterscheiden sich aus ihrer Entstehung heraus nur durch die verwendete Terminologie und ihr Einsatzgebiet.

Das AAF referenziert wie das MXF auf Quellmaterial innerhalb und außerhalb des Containers. Es beschreibt seine verwendeten Inhalte und deren Zusammenstellung ähnlich wie eine Schnittliste (*Edit Decision List*). Darüber hinaus lassen sich umfangreiche Spezialeffekte, Bild- und Tonmanipulationen sowie interaktive Elemente speichern.⁸⁷ Somit können unterschied-

⁸⁷Vgl. Schmidt S. 158

liche non-lineare Bearbeitungsprogramme problemlos an einem Projekt arbeiten oder die kompletten Projektdaten weitergeben. Alle Bearbeitungsparameter werden als Metadaten gespeichert, das Originalmaterial bleibt jedoch unverändert.

Die Spezifikation des Formats ist in der Hand der AMWA, die bis 2007 als *AAF Association* bekannt war. Sie ist eine Vereinigung von Herstellern und Nutzern. Mit dem AAF soll ein plattformübergreifender Austausch von Postproduktionsinhalten realisiert werden. Es werden zudem neue Möglichkeiten in den Bereichen Redaktion, Produktion und Rich Media⁸⁸ Authoring geschaffen.⁸⁹

Struktur Strukturell ist das AAF über weite Teile zu vergleichen mit dem MXF, trotz der höheren Komplexität. Statt Material Packages existieren *MasterMobs*, an Stelle von File Packages gibt es *File SourceMobs* und Tracks heißen *MobSlots*. Über den MasterMobs stehen jedoch noch die *CompositionMobs*, die den Anforderungen an aufwendige Postproduktion gerecht werden.

- **CompositionMobs** beschreiben kreative Entscheidungen, wie Essenz kombiniert oder modifiziert wird. Sie referenzieren auf MasterMobs.
- **MasterMobs** sammeln und synchronisieren Essenzdaten. Sie geben indirekten Zugriff zur Essenz unabhängig von den Details des Speichers.
- **File SourceMobs** geben indirekten Zugriff auf die Essenz und beschreiben die Essenz, die als Computerdateien gespeichert sind.
- **Physical SourceMobs** beschreiben physische Medien wie ein Tape.
- **MobSlots** sind äquivalent zu MXF Tracks, z.B. V1, A1, A2.
 - **Static MobSlots** beschreiben Essenz ohne Zeitbedingung, z.B. Text.
 - **Timeline MobSlots** sind für Essenz mit fester oder kontinuierlicher Zeitbedingung (Video, Audio, Timecode).

⁸⁸Rich Media: audiovisuelle Internet-Inhalte wie z.B. Flash

⁸⁹Vgl. Wells S. 45

2 Grundlagen des dateibasierten Arbeitens

- **Event MobSlots** finden Einsatz bei Essenz mit irregulärer Zeitbedingung, z.B. GPI-Signale⁹⁰.

⁹⁰GPI - General Purpose Interface, vielfältig einsetzbare Schnittstelle für Steuersignale

3 Dateibasierte Aufzeichnung mit Ingex

Die direkte Aufzeichnung von Broadcast Audio- und Videosignalen auf Servern oder in serverbasierende, vernetzte Schnittumgebungen ist die Grundlage für effiziente Postproduktions-Workflows.⁹¹ Sie umgeht die Zwischenspeicherung des Materials auf optischen oder bandbasierten Datenträgern und spart damit erheblich Zeit. Solche Ingest-Server bieten oft große Funktionalität, warten jedoch mit hohen Kosten auf. Daher sind leistungsfähige Systeme gefordert, die als Schnittstelle zum dateibasierten Arbeiten dienen und trotzdem mit geringen Budgets umzusetzen sind. In diesem Kapitel soll daher die Open-Source-Software *Ingex* beschrieben werden, mit der multiple digitale Videosignale (SDI) auf Basis von Standard-IT aufgezeichnet werden können.

Ingex wurde von der *British Broadcasting Corporation* (BBC) entwickelt und steht unter der GNU *General Public License*⁹² als Open-Source-Software für Linux-Distributionen zur freien Verfügung. Unter der Unterbezeichnung *Ingex Studio*, das hier im Mittelpunkt stehen soll, verbirgt sich ein Mehrkamera-Aufzeichnungssystem. Mit PC-Komponenten „von der Stange“ und leistungsfähigen Capture-Karten lassen sich HD-Signale auf Festplatten aufzeichnen und sofort für Schnittprogramme bereitstellen. Ursprünglich für die Fernsehstudio-Umgebung entwickelt, habe ich das System für die Außenübertragung adaptiert (genauer dazu in Kapitel 4 und 5).

Interoperable Lösungen basieren auf offenen Dateiformaten für das Material und die damit verbundene Metadaten.⁹³ Daher ist *Ingex* rund um die Formate MXF und AAF gebaut. Sie ermöglichen den schnellen Austausch von AV-Daten und beschreibenden Informationen. So lassen sich beispiels-

⁹¹Vgl. BBC: White Paper (WHP) 133 S. 3

⁹²GNU GPL: spezielle Lizenzierung für freie Software, die auch kommerzielle Nutzung erlaubt

⁹³Vgl. WHP133 S. 3

weise Multicam-Sequenzen und Schnittlisten per AAF in das Schnittsystem übertragen.

Im Folgenden wird auf den Entwicklungshintergrund dieses Ingestsystems eingegangen, mit Blick auf die Besonderheiten von Open-Source-Software. Der modulare Aufbau von Inges wird dargelegt und Formate, Codecs, Tools sowie seine Einsatzgebiete beschrieben.

3.1 Entstehung von Inges als Werkzeug zur automatisierten Studioproduktion

Mit dem Umstieg auf dateibasiertes Arbeiten ergab sich die Möglichkeit, Kosten und Zeit einzusparen. Die Abteilung *Research & Development* der BBC wollte daher gegen 2005 bei kostengünstig produzierten Fernsehformaten die praktischen Workflows im Studiobereich verbessern. Einzelne Feeds wurden bisher auf Bändern aufgezeichnet und mussten für den non-linearen Schnitt zeitaufwendig überspielt werden.

Offene, standardisierte Dateiformate wie MXF und AAF sollten behilflich sein, Standard-IT in bestehende Produktionswerkzeuge zu integrieren.⁹⁴ Effiziente dateibasierte Workflows können so mit vergleichsweise geringen Kosten erreicht werden.⁹⁵ Möglich wurde das durch fallende Anschaffungskosten und steigende Rechenleistung von PC-Hardware.

Die Hardware-Kosten eines vollständigen Inges-Systems belaufen sich auf 5.000-10.000 Euro, je nach Anzahl der HD/SD-Aufzeichnungskanäle und sonstiger Ausstattung. Die Servertechnik lässt sich schon ab 1.000 Euro realisieren, ein internes Software-RAID ab 500 Euro. Die teuerste Komponente ist derzeit mit rund 3.500 Euro die Capture-Karte. Bis auf diese Karten kann komplett auf Computertechnik „von der Stange“ zurückgegriffen werden. Die Software Inges Studio und das notwendige GNU/Linux-Betriebssystem sind frei verfügbar. Die dort entstehenden Kosten sind auf das Bedien- und Wartungspersonal beschränkt.

Als Pilotprojekt diente die BBC-Kindersendung *BAMZOOKi*. Zwei Staffeln der Spielshow wurden bis Ende 2005 mit einem Inges-Prototypen aufgezeichnet. Das Studioformat mit aufwendiger Postproduktion durch Computeranimationen war für den Einsatz eines dateibasierten Ingestservers

⁹⁴Vgl. WHP133 S. 5

⁹⁵Vgl. BBC: White Paper (WHP) 141 S. 3

3 Dateibasierte Aufzeichnung mit Ingeg

prädestiniert. Neben dem Programmbild mussten mehrere Kamerasignale separat aufgenommen werden. Gleichzeitig sollten die Notizen der Produktionsassistenten (PA) mitgezeichnet und für das Schnittsystem zugänglich gemacht werden.⁹⁶ Die PA schrieb Start- und End-Timecode jeder Aufnahme auf und kommentiert, ob das Take gut oder schlecht war.

Eine direkte serverbasierte Aufzeichnung spart die Zeit des Bändereinspiels und des manuellen Eingebens der Metadaten. Die Postproduktion kann sofort nach der Aufzeichnung beginnen. Seit dem Pilotprojekt arbeitete die BBC kontinuierlich an einer verbesserten Interoperabilität mit professionellen Schnittsystemen. Es besteht die Möglichkeit, MXF OP-Atom zu erstellen, welche nativ im Avid Media Composer bearbeitet werden. Zusätzlich kann Ingeg mittels QuickTime-Container Material für Final-Cut-Systeme bereitstellen.

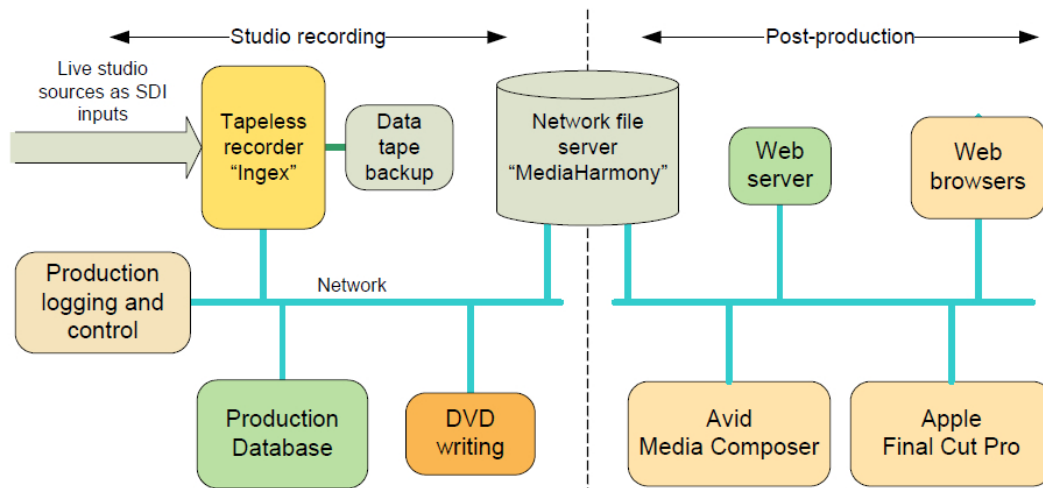


Abbildung 3.1: Ingeg im System eines vernetzten Studios

Abbildung 3.1 zeigt die vernetzte Studioumgebung. Der Recorder ist das Herzstück. Er zeichnet die Live-Quellen auf Festplatte. Er ist direkt mit einem Dateiserver für Mediendaten verbunden, der das aufgezeichnete Material für die Postproduktion zur Verfügung stellt. Gleichzeitig ergeben sich Möglichkeiten, Metadaten in die Datenbank einzutragen und per Weboberfläche das Material zu verwalten.

Ursprünglich zeichnete die BBC für SDTV auf. Dazu verwendete man *Motion-JPEG*⁹⁷ 2:1 (online), *Motion-JPEG* 15:1 (offline) und lineares *PCM*⁹⁸,

⁹⁶Vgl. WHP133 S. 5

⁹⁷intraframe-codierte Videodatenreduktion auf Basis von JPEG

⁹⁸Pulscodemodulation: Modulationsverfahren für digitalen Ton, beschreibt in der Regel nicht datenreduzierten Ton

um hohe Qualität in der Nachbearbeitung zu sichern. 2:1 MJPEG (mit ca. 70 MBit/s) lag dabei laut Messungen um rund +4 dB PSNR⁹⁹ über DigitalBeta-cam-Material.¹⁰⁰ Mittlerweile ist das IngeX-System vollständig HD-fähig und zahlreiche SD- und HD-Codecs sind implementiert worden. Das System wurde für anspruchsvollere Aufgaben weiterentwickelt, um beispielsweise auch für Drama-Serien eingesetzt zu werden.

3.2 Migration von SD zu HD

Die Komponenten von IngeX ermöglichen einen unproblematischen Übergang von *Standard Definition Television* (SDTV) zu *High Definition Television* (HDTV).¹⁰¹ Für die dateibasierte HD-Produktion sind eine HD-fähige SDI-I/O-Karte und zusätzliche PC-Rechenleistung für die Echtzeit-Enkodierung die wesentlichen Grundlagen. IngeX ist in der Lage, in Echtzeit *DVCPRO HD*, *DNxHD* (SMPTE VC-3), *XDCAM HD 422* und unkomprimiertes HD¹⁰² zu enkodieren, sowie als MXF oder QuickTime zu wrappen.

Gigabit-Ethernet und Speichernetzwerke bieten die Infrastruktur für die Aufzeichnung in HD-fähige Schnittumgebungen. HD erfordert mit 100 Mbit/s (DVCPRO HD) bis 185 Mbit/s (DNxHD 185) pro Kanal signifikant mehr Speichertiefe und höhere Schreibgeschwindigkeiten der Festplattensysteme. SD-Online-Codecs liegen mit 50 Mbit/s (DVCPRO 50, IMX 50) bis 70 Mbit/s (MJPEG 2:1) deutlich darunter.

Speichersysteme müssen daher um den Faktor 2-3 erweitert werden. Komprimierte HD-Inhalte über Gigabit-Netzwerke zu schneiden ist möglich.¹⁰³ 10-Gigabit-Netzwerke sind jedoch für spezielle Anwendungsfälle nötig (siehe 5.5 auf S. 77).

⁹⁹ PSNR: Spitzen-Signal-Rausch-Verhältnis, Maß für verlustbehaftete Komprimierung

¹⁰⁰ Vgl. WHP133 S. 8

¹⁰¹ Vgl. WHP155 S. 11

¹⁰² Ebd. S. 11

¹⁰³ Ebd. S. 11

3.3 Besonderheiten der Open-Source-Entwicklung

Die Weiterentwicklung von bestehender Software ist ein Eckpfeiler von Software-Konstruktion. Bei Implementation, Tests und Fehlerbehebung kann von reduziertem Zeitaufwand profitiert werden. Ein Videoserver erfordert zahlreiche Video- und Audiocodecs, plattformübergreifende Netzwirkommunikation und mit Zielsystemen kompatible Dateiformate. Diese Anforderungen ließen sich für die BBC am besten durch den Einsatz von *Open-Source-Software* (OSS) erfüllen.¹⁰⁴ Nur so konnte sich die Entwicklungsarbeit auf die wesentlichen Komponenten von Ingeg konzentrieren.

Eine erste Ingeg-Version von 2006 konnte so zu 99,4% auf bereits vorhandenen Quellcode zurückgreifen.¹⁰⁵ Die wichtigsten genutzten OSS-Komponenten sind im Folgenden gelistet:

- **ACE** - Plattformübergreifende CORBA-Implementation zur Kommunikation der Recorder
- **AAF Toolkit** - Lesen und Schreiben von MXF- und AAF-Dateien
- **FFmpeg** - Implementation zahlreicher Codecs und Containerformate

Erst die Verfügbarkeit dieser Software-Bibliotheken ermöglichte es dem kleinen BBC-Entwicklerteam, in kurzer Zeit einen eigenen Ingeg-Server aufzubauen. Dadurch, dass Ingeg selbst wiederum Open-Source verfügbar gemacht worden ist, können die Anwender direkt an der Entwicklung teilhaben und die Workflows verbessern. Offene Standard-Dateiformate geben hierbei die Schnittstelle zu kommerziellen Anwendungen wie dem Avid Media Composer.¹⁰⁶

Software, die offen entwickelt wird, bringt auch Nachteile mit sich. Die Weiterentwicklung ist immer von Programmierern und Unterstützern abhängig, die keiner kommerziellen Bindung unterliegen. Im Falle von Ingeg leitet die BBC das Projekt. Die BBC arbeitet jedoch mit einem kleinen Team und ist daher auf gemeinschaftliches Arbeiten und den Austausch mit Nutzern angewiesen. Ausführliche Tests, Fehlerbehebung und Feedback sind not-

¹⁰⁴Vgl. WHP133 S. 11

¹⁰⁵Ebd. S. 11

¹⁰⁶Ebd. S. 12

wendig. Ein OSS-Projekt kann schnell zum Stillstand kommen, wenn das Interesse an der Software schwindet.

Außerdem ist Ingex ein Broadcast-Werkzeug, welches in dauerhafter Entwicklung ist. Zwar ist die Veröffentlichung einer ersten stabilen Version geplant¹⁰⁷, es handelt sich jedoch um kein fertiges Produkt, das kommerziell vertrieben werden soll. Daher müssen Zugeständnisse an den Umgang mit der Software gemacht werden.

Zum Beispiel ist die Installation aufwendig und kompliziert. Es ist viel computertechnisches „Know-How“ erforderlich, wenn ein Ingex-System eingerichtet werden soll. Im Gegensatz zu kommerziellen Produkten kann keine Serviceleistung eingefordert werden. Die BBC-Entwickler zeigen sich zwar sehr entgegenkommend bei Hilfestellungen zur Software, sie basieren jedoch auf Freiwilligkeit.

3.4 Modularer Aufbau

Ingex ist modular aufgebaut. Der Anwender kann die Software auf seine Bedürfnisse zuschneiden. Dazu kann er einzelne Komponenten aktivieren oder deaktivieren, bearbeiten und flexibel einsetzen. Die wichtigsten Module erläutere ich kurz, um Funktionsweise und Möglichkeiten von Ingex darzulegen.

3.4.1 Capturing

Im Umfeld von Fernsehproduktionen sind digitale Videosignale unter dem Standard SDI (*Serial Digital Interface*) üblich. Um diese aufzuzeichnen, sind spezielle Capture-Karten¹⁰⁸ mit SDI-Eingängen nötig. Sie lesen die SDI-Rohdaten und schreiben sie in einen gemeinsam mit dem Ingest-System genutzten Pufferspeicher.

Der Speicher hat eine Ringform, d.h. ab einer gewissen Anzahl von gepufferten Frames (in der Regel mindestens 5 Sekunden) werden alte SDI-Daten wieder überschrieben. Dieser sogenannte *Shared Memory Ring Buffer* (Abbildung 3.2) gibt dem Encoder ein Zeitfenster, um die Daten zu lesen und die Codierung der Bild- und Tonsignale zu errechnen. Für jedes Ein-

¹⁰⁷Auskunft von Philip de Nier, BBC-Entwickler, 10.07.2011

¹⁰⁸vom Engl. „Capture“ - einfangen, erfassen

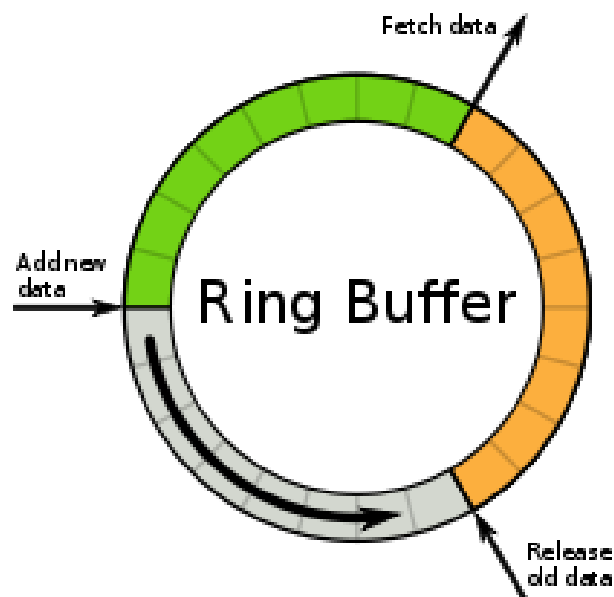


Abbildung 3.2: Der Ring Buffer speichert die eingehenden Daten zwischen (Quelle: Wikipedia)

gangssignal gibt es einen separaten Ring. IngeX stellt den entsprechenden Capture-Prozess zur Verfügung, der den Puffer steuert.

Die Software IngeX ist momentan nur auf SDI-Capture-Karten der Firma DVS¹⁰⁹ ausgerichtet. An der Unterstützung anderer Karten wird gearbeitet. Der Quellcode muss auf die jeweilige SDI-Hardware angepasst werden. Durch den modularen Aufbau von IngeX ist der Einsatz alternativer Karten theoretisch problemlos implementierbar. Voraussetzung sind: ¹¹⁰

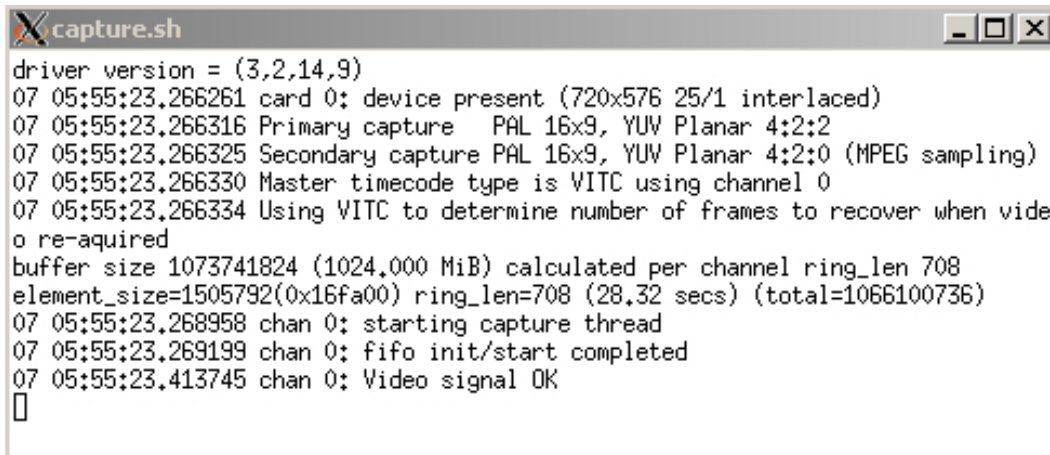
- die Verfügbarkeit von Linux-Treibern
- die Zuverlässigkeit der Karte und ihrer Treiber
- die Kompatibilität der Karte mit dem Motherboard
- die Fähigkeit, Timecodes zu interpretieren

Im Entwicklungsprozess konnten einige Karten nicht verwendet werden, weil sie keine Linux-Treiber unterstützten oder nur für bestimmte Motherboards zugeschnitten waren. Da mit IngeX die Aufzeichnung von HD-Signalen möglich ist, sollte die Karte außerdem HD-SDI-fähig sein.

¹⁰⁹Digital Video Systems AG, Hersteller von Broadcast-Hardware

¹¹⁰Vgl. <http://sourceforge.net/projects/ingex/forums/forum/531547/topic/2725503> (abgerufen am 20.06.2011)

3 Dateibasierte Aufzeichnung mit IngeX

A screenshot of a terminal window titled 'capture.sh'. The window displays a series of log messages from the IngeX capture process. The messages include driver version information, device detection details (720x576 25/1 interlaced), capture parameters (PAL 16x9, YUV Planar 4:2:2), secondary capture settings (MPEG sampling), master timecode type (VITC using channel 0), and buffer size calculations (1024,000 MiB). It also shows the start of the capture thread, FIFO initialization completion, and a video signal OK message.

```
capture.sh
driver version = (3,2,14,9)
07 05:55:23.266261 card 0: device present (720x576 25/1 interlaced)
07 05:55:23.266316 Primary capture PAL 16x9, YUV Planar 4:2:2
07 05:55:23.266325 Secondary capture PAL 16x9, YUV Planar 4:2:0 (MPEG sampling)
07 05:55:23.266330 Master timecode type is VITC using channel 0
07 05:55:23.266334 Using VITC to determine number of frames to recover when video re-acquired
buffer size 1073741824 (1024,000 MiB) calculated per channel ring_len 708
element_size=1505792(0x16fa00) ring_len=708 (28,32 secs) (total=1066100736)
07 05:55:23.268958 chan 0: starting capture thread
07 05:55:23.269199 chan 0: fifo init/start completed
07 05:55:23.413745 chan 0: Video signal OK
█
```

Abbildung 3.3: Ausgabe des Capture-Prozesses von IngeX (Quelle: Screenshot auf IngeX-PC)

Abbildung 3.3 zeigt das Kontrollfenster für das Capturing eines Kanals in SD. Die vorkonfigurierte Auflösung, Bildwiederholrate und TV-Norm des gewünschten Signals sind gelistet. Wenn das anliegende Signal dem entspricht, beginnt die Karte automatisch, Daten in den angegebenen Puffer (hier 1024 Megabyte) einzulesen. In dem Fall dient der VITC des ersten Aufzeichnungskanals als Master Timecode.

Die Fähigkeit Timecodes zu lesen ist bedeutsam für die Postproduktion. Gerade bei Multikamera-Mitschnitten müssen Signale framegenau synchronisiert werden. Timecodes dienen zudem der Orientierung im Material, dem Auffinden bestimmter Szenen und der Zuordnung von Metadaten. IngeX unterstützt folgende Modi, um Timecodes aufzuzeichnen:

- **SYS** - der Timecode synchronisiert sich mit der Systemzeit (Computer)
- **LTC** - der Timecode wird als seriell analoges Audiosignal (meist über XLR) der Karte übergeben - steht für Longitudinal Timecode
- **VITC** - meint hier den VITC in den entsprechenden Zeilen der vertikalen Austastlücke des digitalen Videosignal (eigentlich DVITC) - steht für Vertical Interval Timecode
- **DLTC, DVITC** - stehen hier für die Angaben aus dem Ancillary Timecode (ATC), die als digitale Zusatzdaten eingebettet sind¹¹¹

¹¹¹Vgl. SMPTE RP-188 (überholt) und SMPTE 12M-2-2008

Die Begrifflichkeiten sind hier leider nicht klar abgegrenzt. DVITC beschreibt beispielsweise den dem analogen Signal nachempfundenen VITC im digitalen Signal, hier jedoch den ATC-VITC. Der digitale VITC ist für HD-Signale nicht mehr spezifiziert, daher sollte dort ATC eingesetzt werden.¹¹²

3.4.2 FFmpeg

Das Echtzeit-Encoding der gepufferten SDI-Signale findet mittels der Encodingsoftware *FFmpeg* statt. FFmpeg ist ein Open-Source-Projekt, das sich aus mehreren Programmen zur Aufzeichnung, Konvertierung und zum Streaming von Audio- und Videomaterial zusammensetzt. Die dazugehörige freie Bibliothek libavcodec implementiert zahlreiche Encoder und Decoder. Gängige Containerformate stellt die FFmpeg-Bibliothek libavformat bereit¹¹³, so u.a. auch den Quicktime-MOV-, DV, MPG- und MP-4-Container.

Ingex nutzt eine leicht abgewandelte Version von FFmpeg 0.5, die erweiterte Funktionen zum DNxHD-Encoding bietet. Außerdem beauftragte die BBC einen freien Entwickler, der den *DVCPRO HD* Codec implementierte. Laut Ingex-Entwickler Stuart Cunningham liefert dieser bessere Ergebnisse als die meisten Hardware-Dekoder.¹¹⁴

FFmpeg hat sich aufgrund interner Unstimmigkeiten im März 2011 aufgesplittet in die beiden Projekte *Libav* und *ffmpeg*. Es bleibt abzuwarten, welches Projekt sich künftig durchsetzt.¹¹⁵

3.4.3 Writeavidmxf

Das Modul *Writeavidmxf* ist verantwortlich für das Wrapping von MXF-Dateien. Eine Essenz wird dabei in einen Container gepackt, der dem Specialized Operational Pattern OP-Atom entspricht. Wie der Name schon andeutet, sind die erzeugten Files Avid-kompatibel. Sie lassen sich ohne Importvorgänge oder Re-Wrapping mit einem Avid-Schnittsystem bearbeiten.

Um einen Video-Datenstrom als MXF zu wrappen, ist eine Software-Bibliothek nötig, die einen Stream als Eingangssignal verarbeitet. Dafür hat die

¹¹²Vgl. Alpermann+Velte: Ancillary Time Code (ATC) according to SMPTE 12M-2, unter: http://www.alpermann-velte.com/faq_e/e_atc.html (abgerufen am 20.06.2011)

¹¹³Vgl. Chantelau/Brothuhn: Technologien zur Entwicklung multimedialer Client-Server-Systeme S. 31

¹¹⁴Vgl. Gerloff S. 5

¹¹⁵Vgl. <http://www.pro-linux.de/news/1/16813/libav-soll-ffmpeg-ersetzen.html> (abgerufen am 19.06.2011)

BBC die Bibliothek *libMXF* entwickelt.¹¹⁶ Sie ist in der Lage, Video- und Audioessenz direkt in MXF-Dateien zu schreiben. Wenn OP-Atom-Dateien auf ein für Schnittsysteme freigegebenes Netzlaufwerk geschrieben werden, können diese sofort nach Abschluss der Aufzeichnung bearbeitet und dargestellt werden. Wege, mit dem Schnittsystem während der Aufzeichnung auf die Daten zuzugreifen, beschreibe ich unter Kapitel 5.1.

Writeavidmxf ist normalerweise fester Bestandteil eines Ingests von Op-Atom-Dateien mit Ingex. Das Modul lässt sich aber auch separat per Kommandozeile nutzen. So lassen sich vorliegende Essenzen nach Avid-Standards als MXF wrappen. Writeavidmxf-Files sind zudem kompatibel mit servergestützten Schnittsystemen der Firma Avid (u.a. Interplay, ISIS).¹¹⁷

3.4.4 Recorder

Der Recorder ist ein umfassendes Modul, dass sich verschiedener Programmteile wie FFmpeg und Writeavidmxf bedient. Er ist als *direct-to-disk* konzipiert, für die direkte Speicherung der Daten auf Festplatte. Der Recorder hat die Aufgabe, Signale in voller Qualität zu capturen, zu enkodieren und in Standard-Dateiformaten wie MXF und QT zu speichern.¹¹⁸ Durch die Fortschritte der Rechentechnik lassen sich mit Ingex pro Recorder simultan vier Kanäle in mehreren Qualitätsstufen (Online, Offline, Review) aufzeichnen. Review steht hier für die Materialsichtung, z.B. im MPEG-2-Format.

Der Recorder läuft auf einem *CORBA*-Server (siehe 4.4 auf S. 64). Über den *Naming Service* können mehrere Recorder in einer grafischen Oberfläche (GUI) zusammengefasst und gesteuert werden. Die Anzahl der Aufzeichnungskanäle lässt sich somit über das Kaskadieren von Recordern und Recorder-PCs theoretisch beliebig erweitern. Auf einem PC können auch mehrere Recorder laufen. So lassen sich für bestimmte Anwendungen die Kanäle splitten. Zum Beispiel lässt sich ein Recorder konfigurieren, der drei Kanäle in DVCPRO-HD sowie MPEG-2 aufzeichnet und einen Kanal nur MPEG-2. Beliebige Konfigurationen sind möglich.

Pro Recorder können jeweils drei Aufzeichnungsformate konfiguriert werden. Sofern genügend Rechenleistung vorhanden ist, können mit jedem Recorder zusätzliche Formate aufgezeichnet werden (Abbildung 3.4).

¹¹⁶Vgl. WHP155 S. 6

¹¹⁷Vgl. <http://sourceforge.net/projects/ingex/forums/forum/531547/topic/4375278> (abgerufen am 20.06.2011)

¹¹⁸Vgl. BBC: White Paper (WHP) 186 S. 5

3 Dateibasierte Aufzeichnung mit Ingex

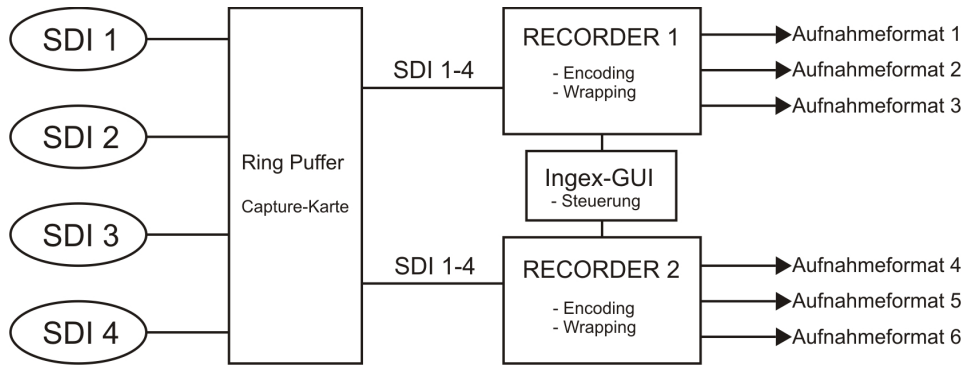


Abbildung 3.4: Schematische Darstellung einer Aufzeichnung von 4 SDI-Quellen mit 2 Recorders (Quelle: eigene Darstellung)

Auf eine START RECORD Anfrage startet der Recorder verschiedene Prozesse, die folgende Aufgaben haben: ¹¹⁹

- Das Schreiben von (un-)komprimierten Video- und Audiodaten für jede Videoquelle.
- Das Enkodieren in weiteren Formaten.
- Die Quellen synchronisieren für die Multicam-Aufzeichnungen und im Quad-Split zur Vorschau bereitstellen.
- Alle Enkodiervorgänge managen.

```

run_recorder2.sh
Main thread
Recorder name "MpegRecorder"

Connected to shared memory.
corbaloc:iiop:1.0@192.168.0.135:8888/NameService - object advertised
Recorder running...
Read settings for config "MpegKonfig"
Updated sources for recorder "MpegRecorder"
06:26:51.500 IngexRecorderImpl::Start(), tc 13:24:35:15, pre-roll 10
CopyManager failed to connect to xferserver.
06:26:51.502 start_record_thread(input0 index 0, start_tc=13:24:35:05 MPEG2 fo
DVD with BITC)
[mpeg2video @ 0x9b80590]rc buffer underflow
06:31:38.888 IngexRecorderImpl::Stop(), tc 13:29:22:24, post-roll 10
06:31:39.185 input0 index 0 duration 7204 reached (total=7204 written=7204 dro
pped=0)
Recording completed ok.
  
```

Abbildung 3.5: Das Ausgabefenster eines Recorders nach einer abgeschlossenen Aufnahme (Quelle: Screenshot auf Ingex-PC)

¹¹⁹Vgl. WHP133 S. 7

3 Dateibasierte Aufzeichnung mit Ingex

Das Recording läuft bis zur STOP Anfrage kontinuierlich. Ab dann werden nur noch die restlichen im Puffer befindlichen Frames enkodiert und geschrieben.

3.4.5 Ingexgui

Die einzelnen Recorder und Aufnahmeprozesse werden in einer grafischen Oberfläche gesteuert, einem GUI (*Graphical User Interface*). Das Modul *Ingexgui* fasst die notwendigen Elemente in einer einfachen, klaren Optik zusammen. Es gibt die Möglichkeit, Recorder zu konfigurieren, Aufnahmen zu starten, aufgezeichnetes Material über einen Player abzuspielen und zahlreiche weitere Einstell- und Kontrollmöglichkeiten.

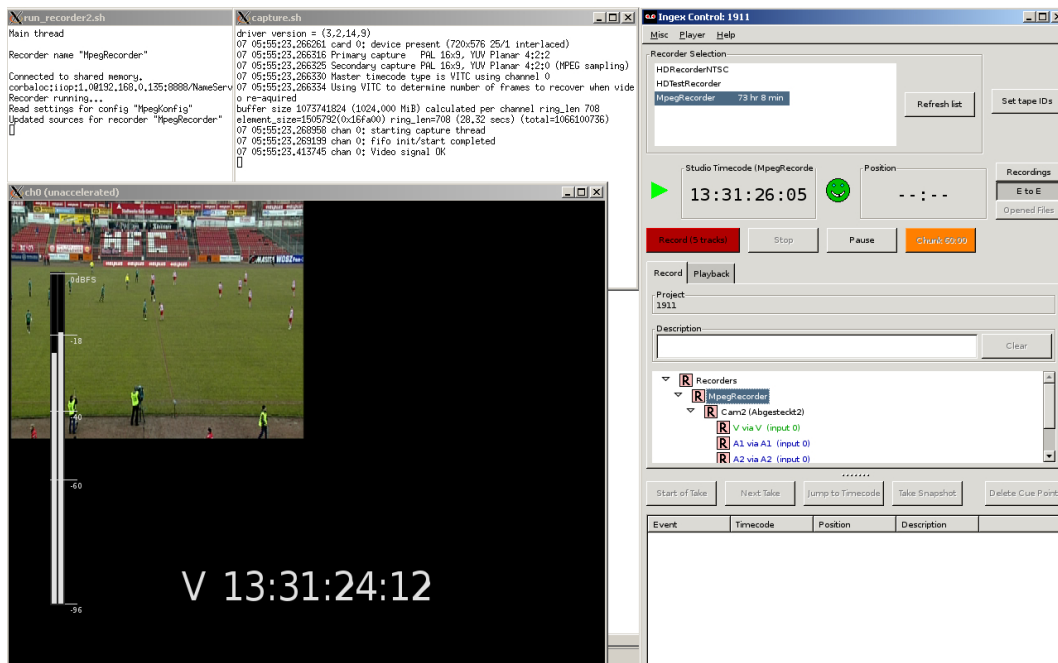
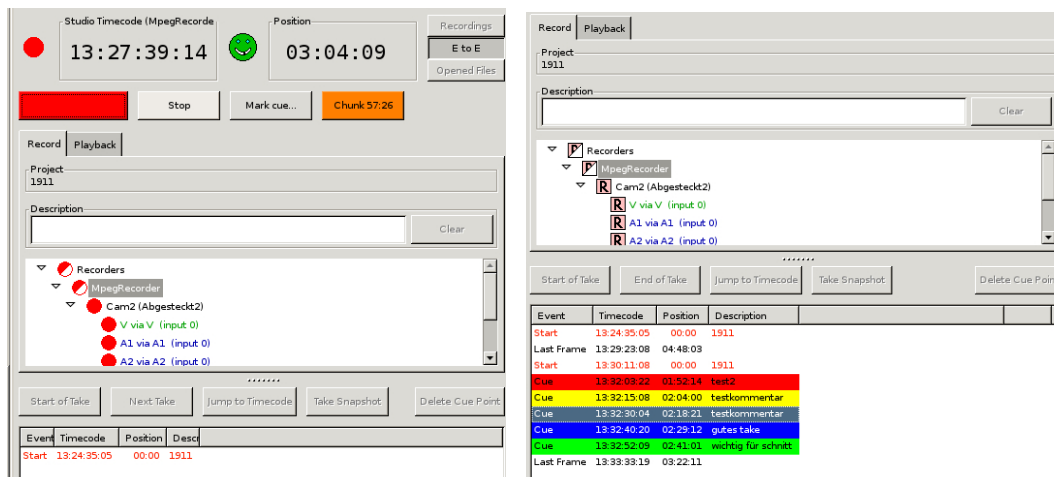


Abbildung 3.6: Ingex-GUI mit Bedienoberfläche, Quadsplit, Capture- und Recorderprozess (Quelle: Screenshot auf Ingex-PC)

Wie die grafische Oberfläche in einer Desktop-Umgebung aussehen kann, zeigt Abbildung 3.6. Im unteren Bereich des Kontrollfensters (rechts) werden die Eingangssignale angewählt und bestimmt, welche Video- und Audiospuren aufgezeichnet werden sollen. Um eine Aufnahme starten zu können, müssen lediglich ein (oder mehrere) Recorder angewählt, ein *Projektname* festgelegt und die *Tape IDs* vergeben werden. Zusätzlich kann man eine Projektbeschreibung (*Description*) eingeben. All diese Metadaten über-

3 Dateibasierte Aufzeichnung mit Ingex



(a) Mit *Mark cue* lassen sich Cue-Punkte setzen (b) Die farbig markierten Cues sind im Schnitt verfügbar

Abbildung 3.7: Ausschnitte der Ingexgui während (a) und nach (b) der Aufnahme (Quelle: Screenshots auf Ingex-PC)

nimmt Ingex in seine MXF- und AAF-Files. Während der Aufnahme lassen sich *Cue-Punkte* setzen (siehe Abbildung 3.7).

Cue-Punkte sind kleine Markierungen oder Hinweise. Sie werden als Kommentar an bestimmte Stellen des Materials gesetzt, zum Beispiel weil eine Szene wichtig war, technische Fehler vorkamen oder um Material allgemein schnell finden zu können. Die in der Ingexgui gesetzten Cues sind in Avid-Schnittsystemen mit identischer Farbe und entsprechendem Kommentar als Locator wiederzufinden. Sie können per AAF importiert werden. Damit kann schon während des Ingests das Logging¹²⁰ vorgenommen werden, was den non-linearen Schnitt erheblich beschleunigt.

Im Kontrollfenster der Ingexgui gibt es neben der Metadaten-Eingabe noch zahlreiche weitere nützliche Optionen. Beispielsweise lässt sich ein automatisches Chunking¹²¹ konfigurieren. Damit wird in einem bestimmten Zeitabstand jeweils ein neuer Clip (neue Datei) angelegt. Das Chunking findet während der Aufnahme statt und ist framegenau, d.h. es geht während eines Chunks kein Material verloren. Die neue Datei setzt exakt am letzten Frame der vorherigen Datei an. Chunks können sowohl periodisch, als auch manuell zu bestimmten Timecodes gesetzt werden.

¹²⁰ Sichten und Erfassen des aufgezeichneten Materials für den non-linearen Schnitt

¹²¹ Engl. für „in Stücke schneiden“

3.4.6 Createaaf

Um Ingeg in allen Funktionen zu nutzen, ist das *AAF Toolkit*¹²² zwingend zu installieren. Es ermöglicht die Erstellung von AAF-Dateien. Das notwendige SDK (*Software Development Kit*) wird als Open Source von der AMWA bereitgestellt. Mit dem daraus für Ingeg kompilierten Modul Createaaf lassen sich AAF-Dateien mittels Kommandozeileingabe oder alternativ Web-oberfläche erstellen.

Durch AAF-Dateien können Metadaten sowie referenziertes Bild- und Tonmaterial vom Ingegserver direkt in das Schnittsystem importiert werden. AAFs sind zudem in der Lage, komplexe Informationen zu transportieren, wie ausführlich in Kapitel 2 beschrieben. Neben Multicam-Gruppierungen und Schnittlisten werden bei Ingeg folgende Metadaten standardmäßig unterstützt:

Clip Name, Tape Name, Tape start timecode, Duration, Track number, Description, Comments, Cue Points, Marked IN/OUT points in clip, Clip UMID, Kind of video or audio coding

Alle Metadaten werden automatisch oder direkt in der Ingeg-GUI erstellt. Die MXF-Dateien, die Ingeg erstellt, beinhalten bis auf die IN/OUT-Marken dieselben Metadaten wie die AAFs. Tape name und Tape start timecode sind nicht standardmäßig im MXF-Container enthalten, auch wenn sie von Avid unterstützt werden.¹²³ Ihr Einsatz sichert jedoch das einfache Mischen von bandbasierten und dateibasierten Quellen. Die Vorteile des bandbasierten Arbeitens sollen dadurch möglichst ähnlich übernommen werden, um bewährte Workflows beizubehalten.

Multi-Kamera-Gruppen können erstellt und per AAF direkt im Avid importiert werden.¹²⁴ Die abgesteckten Kameras liegen dadurch bereits synchronisiert als group clip für den sogenannten Multicam-Schnitt bereit.

Im AAF lassen sich auch Sequenzen für die Avid-Timeline abspeichern. Dadurch können beispielsweise die Bildmischer-Entscheidungen aufgezeichnet und als Schnitte in einer Avid-Sequenz importiert werden (siehe dazu Punkt 3.4.8 auf S. 50).¹²⁵

¹²²<http://aaf.sourceforge.net/>

¹²³Vgl. WHP133 S. 9

¹²⁴Vgl. WHP141 S. 11

¹²⁵Ebd. S. 11

3.4.7 Database und WebIngeg

In einer Datenbank (Database) speichert Ingeg die Konfiguration der Signalquellen, die mit dem Recorder verbunden sind. Auf der Weboberfläche *WebIngeg* lassen sich die Einstellungen per Browser bearbeiten. Alle Metadaten einer Aufzeichnung werden in der Datenbank abgelegt, so z.B. der Start-Timecode oder die Dauer des Clips. Die Daten lassen sich dann über den Browser abrufen. Einzelne Clips können dort mittels eines Suchfilters ausgewählt werden (Abbildung 3.8).

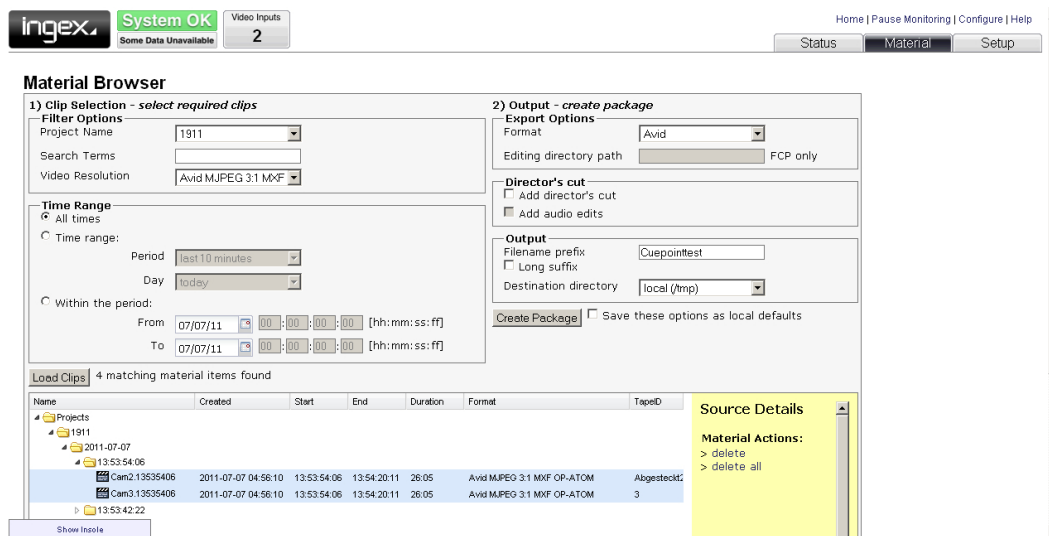


Abbildung 3.8: Die Ingeg-Weboberfläche mit dem Materialbrowser (Quelle: Screenshot auf Ingeg-PC)

Für aufgezeichnete Clips lässt sich unter „Output - create package“ ein AAF-File erzeugen, das einen unkomplizierten Drag&Drop-Import von Medien- und Metadaten in ein Avid-Schnittsystem ermöglicht.¹²⁶ Das AAF kann dabei sowohl für mehrere ausgewählte Clips, als auch für eine Multicam-Gruppe von beispielsweise vier Videoquellen erstellt werden. Die Multicam-Gruppe steht dem Cutter dann eins zu eins im Avid zur Verfügung. Für Final-Cut-Systeme lassen sich die Schnittinformationen zusätzlich als XML exportieren.

Die komplette Recorder-Konfiguration kann in der Weboberfläche vorgenommen werden. Alle Änderungen werden in die zentrale Ingeg-Datenbank geschrieben. Abbildung 3.9 zeigt, wie sich beispielsweise Zielverzeichnis (Encode_Dir) und Encodingformat (Encode_Resolution) auswählen lassen.

¹²⁶Vgl. WHP155 S. 7

3 Dateibasierte Aufzeichnung mit Ingeg

Zugleich findet das Monitoring per Browser statt (Abbildung 3.10). Die laufenden Aufnahmen werden überwacht. Systemauslastung (CPU) und freier Speicherplatz sind grafisch dargestellt und aktualisieren sich sekundlich.

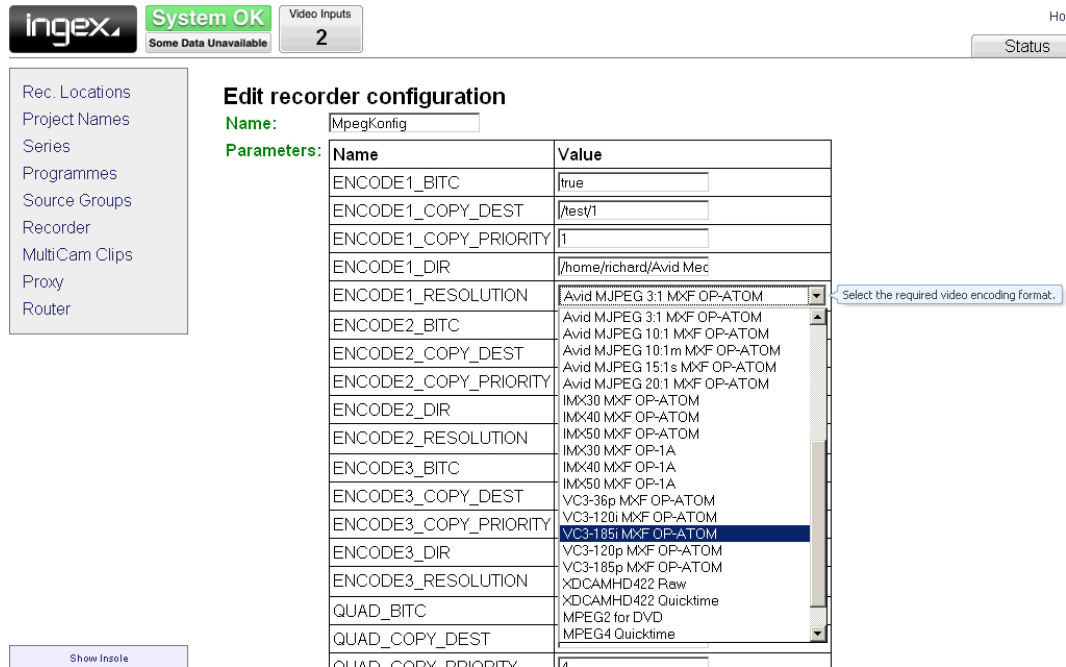


Abbildung 3.9: Recorder-Einstellung in der Weboberfläche (Quelle: Screenshot auf Ingeg-PC)

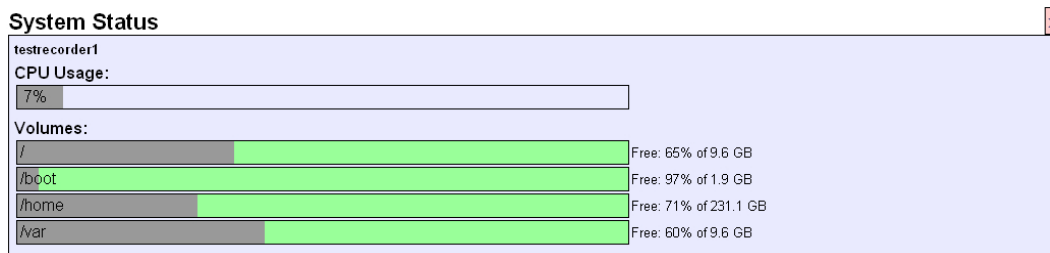


Abbildung 3.10: Die Systemkapazitäten sind grafisch dargestellt (Quelle: Screenshot auf Ingeg-PC)

3.4.8 Directors Cut

Um die Entscheidungen des Bildmischers beim Schnitt des Programmbildes (PGM) zu speichern, kann die jeweilige Bildauswahl als Rohschnitt in einer AAF-Sequenz gespeichert werden. Mit dem sogenannten Directors

3 Dateibasierte Aufzeichnung mit Ingex

Cut Feature (DC) lässt sich das Bildmischer-PGM aufzeichnen. Über sogenanntes *Frame Matching* vergleicht DC das Programmbild jeweils mit den einzelnen Kamerabildern.¹²⁷ Dazu nutzt es eine technisch aus Helligkeit und bestimmten Bildbereichen berechnete Signatur eines jeden Bildes.

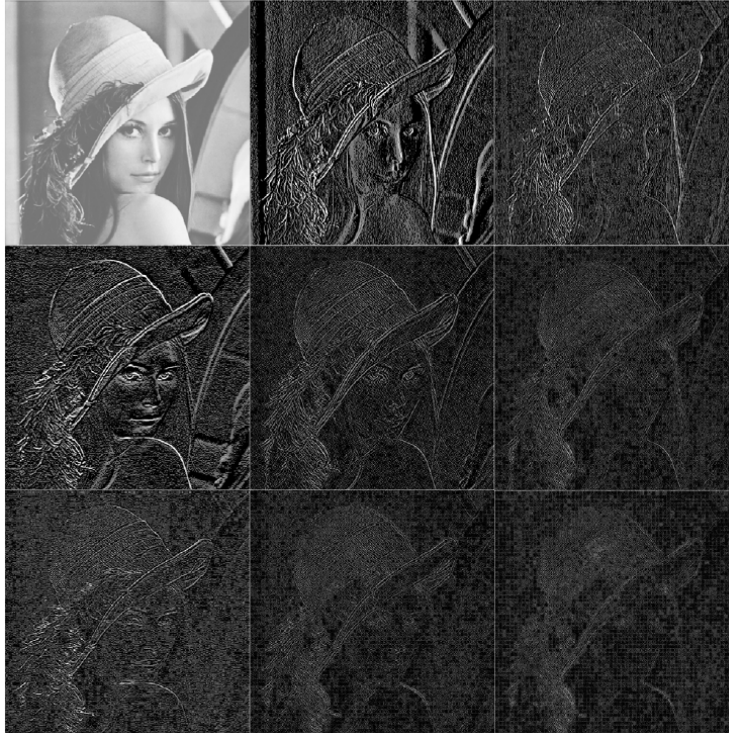


Abbildung 3.11: Directors Cut nutzt Bildtexturmessung (Frame Matching), um Signalquellen abzugleichen (Quelle: BBC White Paper 186)

Die aus dem Vergleich entstandenen Metadaten speichert Ingex im AAF. Die Frame Matching Technik macht das System somit unabhängig von der Bauart des Mischers. Es müssen außerdem nicht alle Quellen aufgezeichnet werden. Unbekannte Bilder werden während des Vergleichs einfach übersprungen.

Dadurch steht der Postproduktion nicht nur ein Rohschnitt zur Verfügung, es sind auch alle abgesteckt aufgezeichneten Kameraquellen in der Sequenz verlinkt. Somit kann der Cutter das Material problemlos in der Zeitachse trimmen. Voraussetzung ist selbstverständlich die Aufzeichnung des Programmbildes (Ausgang des Bildmischers) und der einzelnen Kameras.

Falls die Daten des Bildmischers nicht zur Verfügung stehen, können bis zu 4 Quellen auch in der Ingex-Oberfläche mittels Quad-Splits geschnitten

¹²⁷Vgl. WHP186 S. 9

werden.¹²⁸ Der Rohschnitt steht dann ebenfalls mit den verlinkten Kameraquellen in der Postproduktion zur Verfügung.

3.4.9 Quadsplit

IngeX verfügt über eine Player-Funktion, die sich in der grafischen Oberfläche aufrufen lässt. Aktuell anliegende Signale oder Aufzeichnungen können damit synchron abgespielt und kontrolliert werden. Mit dem Quad-Split-Modul lassen sich 4 Quellen inklusive Timecode gleichzeitig in einem Player-Fenster darstellen. Bildsignal und bis zu 16 Audiospuren werden in Echtzeit angezeigt. Der Quad-Split-Prozess benötigt zwar einiges an Rechenlast, da alle 4 Videoströme zur Voransicht in Video und Audio synchronisiert und dargestellt werden müssen, das sollte jedoch mit einem leistungsfähigen System nicht ins Gewicht fallen.



Abbildung 3.12: Bis zu 4 Videosignale und 16 Audiospuren lassen sich im Quadsplit darstellen (Quelle: Screenshot auf IngeX-PC)

Der Quadsplit basiert auf Modul *IngeXPlayer*, welches als Schnittstelle zwischen dateibasierten Medien und SDI-basiertem Broadcast-Equipment dienen soll. Bildmaterial aus MXF-Dateien lässt sich über den Ausgang der

¹²⁸Vgl. WHP186 S. 9

SDI-I/O-Karte am PC ausspielen. Clips können dadurch parallel am SDI-Ausgang und PC-Monitor angezeigt werden. Zudem ist schneller, zufälliger Zugriff auf das Material möglich, ohne langsames Shutteln. Über den Player wird wechselseitige Kompatibilität mit anderen Broadcast-Geräten hergestellt, zum Beispiel lässt sich so simpel Material mit bis zu 8 eingebetteten Audiospuren und VITC/LTC auf Band ausspielen.¹²⁹

3.5 Unterstützte Formate und Codecs

Inges unterstützt zahlreiche Videokompressions- und Containerformate. Da es für den Austausch mit der Postproduktion konzipiert ist, stützt es sich auf die Container *MXF OP-Atom* und *QuickTime*. Damit ist der reibungslose, native Import in die verbreiteten Schnittsysteme von Avid (MXF) und von Apple (QuickTime) möglich. In den Containern lassen sich eine Reihe gängiger Codecs in SD und HD aufzeichnen. Allen voran die HD-Codecs DNxHD, DVCPRO HD und XDCAM-HD 422. Pro Eingangssignal lassen sich 8 Audiokanäle mitzeichnen.

Einen Überblick soll die folgende Tabelle (3.1) geben. Sie listet eine Auswahl an professionellen Codecs und Containerformaten. Die Aufzeichnungsmöglichkeiten von Inges werden gegenübergestellt mit den nativen Bearbeitungsmöglichkeiten der Schnittsysteme Avid Media Composer (3.5 bis 5.0) und Final Cut Pro (Version 7). In der rechten Spalte wird zudem ein anderes Aufzeichnungssystem zum Vergleich herangezogen, die EVS XT2.¹³⁰

Es wird deutlich, dass für die Schnittsysteme jeweils bestimmte Formate in Frage kommen. Im HD-Bereich bieten sich für Avid-Clients DNxHD und DVCPRO HD im OP-Atom-Container an. Final Cut Pro kann mit DVCPRO HD und XDCAM-HD 422 im QT-Format bedient werden.

Für SD-Aufzeichnungen kann in DVCPRO 50 und IMX 50 angeliefert werden. Nicht gelistet, aber ebenfalls verfügbar sind Motion-JPEG von den Qualitätsstufen 2:1 bis 20:1, DVCPRO 25 sowie IMX 30 und 40. Eine Besonderheit ist die Codierung in MPEG-4 und MPEG-2, die parallel zur normalen Aufzeichnung stattfinden kann. Damit lassen sich schnell verfügbare Clips mit geringer Dateigröße für Redakteure mitzeichnen.

Nicht unterstützt werden AVC-Intra und ProRes. Eine Implementation ist

¹²⁹Vgl. WHP155 S. 10

¹³⁰EVS XT2: Videoserver der Firma EVS Broadcast Equipment, weit verbreitet in der Fernseh-Außenübertragung

3 Dateibasierte Aufzeichnung mit IngeX

Codec	Container	IngeX	AVID Media Composer (nativ)	Final Cut Pro (nativ)	EVS
Panasonic					
DVCPRO-50	MXF OP-AT	ja	ja	nein	ja
	MXF OP-1A	nein	ab 5.0	nein	ja
	QT	ja	ab 5.0	ja	ja
DVCPRO-HD	MXF OP-AT	ja	ja	nein	ja
	MXF OP-1A	nein	ab 5.0	nein	ja
	QT	ja	ab 5.0	ja	ja
AVC-Intra 100	MXF OP-AT	nein	ab 4.0	nein	ab XT3/2+
	MXF OP-1A	nein	ab 5.0	nein	ab XT3/2+
Sony					
IMX50	MXF OP-AT	ja	ja	nein	ja
	MXF OP-1A	ja	ab 5.0	nein	ja
XDCAM-HD 422	QT	ja	ab 5.0	ja	nein
Avid					
DNxHD 36	MXF OP-AT	ja	ja	nein	nein
	MXF OP-1A	nein	ab 5.0	nein	nein
DNxHD 120	MXF OP-AT	ja	ja	nein	ja
	MXF OP-1A	nein	ab 5.0	nein	ja
DNxHD 185	MXF OP-AT	ja	ja	nein	ja
	MXF OP-1A	nein	ab 5.0	nein	ja
Apple					
ProRes 422 120	QT	nein	ab 5.0	ja	ja
ProRes 422 185	QT	nein	ab 5.0	ja	ja
Andere					
MPEG-4	QT	ja	ab 5.0	ja	
MPEG-2 DVD	MPG	ja	k.A.	k.A.	

Tabelle 3.1: Aufzeichnungs- und Schnittsysteme im Vergleich (Quelle: eigene Darstellung)

3 Dateibasierte Aufzeichnung mit Ingeg

zum jetzigen Stand nicht vorhanden. Mit den entsprechenden, lizenzierten SDKs dürften sich die Codecs jedoch von einem Programmierer in die Ingeg-Software integrieren lassen. Vom proprietären Vergleichssystem EVS XT2/3 werden sie bereits unterstützt. Das Wrapping von OP1a-Containern ist mit EVS-Software möglich. Ingeg ermöglicht Aufzeichnung im OP1a-Container momentan nur für IMX50. Für XDCAM-HD 422 wird OP1a demnächst implementiert, was im Bezug auf XDCAM-Umgebungen interessant erscheint. Eine Übersicht über alle verfügbaren Codecs¹³¹ befindet sich im Anhang.

MPEG-2 Parallel zum hochqualitativen Ingeg lassen sich MPEG-2 kodierte MPG-Dateien aufzeichnen. Diese sind ideal geeignet zur Sichtung durch Redakteure oder den Regisseur. Mit Burned-in-Timecode (BITC), einem im Bild aufgezeichneten Timecode, kann das Material szenenweise und Timecode-genau analysiert werden. Das MPEG-2-Format ermöglicht zudem die schnelle Erstellung von DVDs.

TV-Normen Die Verarbeitung von HDTV ist im Zeilensprungverfahren (*interlaced*, i) und mit progressiver Abtastung (*progressive*, p) möglich. International sind die Auflösungen 1920x1080 (Bildpunkte x Zeilen) und 1280x720 standardisiert. Einige Studiogeräte unterstützen progressive Abtastung nicht, daher existiert zusätzlich eine besondere Form der progressiven Abtastung, *progressive segmented Frames* (psF). Dort wird ein Vollbild in zwei bewegungsgleiche Teilbilder zerlegt, die technisch den Halbbildern des Zeilensprungverfahrens entsprechen. Für Europa ergeben sich daher die Abtastformate 1080i/25, 1080p/25, 1080psF/25 und 720p/50, für die USA und Japan 1080i/29,97, 1080p/29,97, 1080psf/29,97 und 720p/59,94.¹³²

Ingeg unterstützt 1080i, 1080psf und 720p für die Bildwechselraten 25, 29,97 und 50 Hz. Im SD-Bereich kann in den üblichen Normen 576i/25 und 480i/29,97 aufgezeichnet werden. Eigene Tests ergaben jedoch, dass nicht alle Codecs/Container vollständig funktionieren, wenn Bildwechselraten mit Drop Frame genutzt werden (29,97, 59,94). Die Aufzeichnung in DVCPRO-HD im MXF OP-Atom ergab für das Abtastformat 1080i/29,97 Fehler, während DNxHD 185i (OP-Atom) und DVCPRO-HD (QT) möglich sind.

¹³¹Vgl. BBC: Ingeg Installation Guide S. 28

¹³²Vgl. BET Fachwörterbuch: HDTV, unter: <http://www.bet.de/Lexikon/Begriffe/hdtv.htm> (abgerufen am 02.07.2011)

3.6 TransferToP2

Der verstärkte Einsatz des MXF hat in vielen Produktionsbereichen die Spanne an Workflows erweitert. Das P2-Format von Panasonic lässt beispielsweise den unkomplizierten Ingest und Schnitt auf Laptops zu. Es zeichnet MXF OP-Atom auf Flashspeichern auf. Das Material lässt sich nativ mit Schnittsystemen wie Avid bearbeiten. Problematisch ist jedoch der Playout von einem Laptop.

Daher stellt Ingest ein Tool bereit, um fertig geschnittene Sequenzen zurück auf die P2-Karte zu spielen. Mit *TransferToP2* werden die Avid-MXF zurück in P2-kompatible MXF-Container gewrappt, die den besonderen Beschränkungen des P2 MXF OP-Atom entsprechen.¹³³ Der Playout ist nach dem dateibasierten Transfer auf die P2-Karte direkt von der Kamera möglich.

Anwendung hat das Tool unter anderem bei der Fußball-WM 2006 gefunden. Cutter von BBC Sport haben ihre fertig geschnittenen Sequenzen vom Laptop zurück auf die P2-Kamera transferiert, von wo aus sie über SDI ausgespielt worden. In der Folge haben auch kommerzielle Produkte diese Lösung für eine verbesserte Interoperabilität genutzt.¹³⁴

¹³³Vgl. WHP141 S. 12

¹³⁴Vgl. WHP155 S. 8

4 Aufbau und Konfiguration eines Ingex-Videoservers auf Debian-Plattform

Während meines Praktikums bei der *Media Mobil GmbH* (MMG) entstand die Zielsetzung, ein dateibasiertes Aufzeichnungssystem auf Basis von Ingex zu entwickeln. Ein System, welches eine funktionierende Schnittstelle vom SDI-basiertem Broadcast-Equipment zur dateibasierten Postproduktion bietet. Als Fernseh-Dienstleister muss die MMG normalerweise auf sehr kostspielige Technik zurückgreifen, um HD-Signale aufzuzeichnen und ohne Umwege in einer non-linearen Schnittumgebung bereitzustellen.

Mit Ingex bot sich daher eine vielversprechende und bezahlbare Möglichkeit, auf Basis von Standard-IT eigenständig HD-Aufzeichnungskanäle zu erzeugen. Ein solches System sollte effizient, einfach und günstig sein, im Bezug auf optimierte HD-Workflows aber auch leistungsfähig, flexibel und erweiterbar. Insofern mussten die vorhandenen Ressourcen effektiv genutzt werden. Die Ingest-Lösung sollte zudem an die speziellen Anforderungen in der Außenübertragung angepasst werden.

Über rund vier Monate arbeitete ich daran, die Ingex-Software an die Linux-Distribution *Debian* anzupassen, einen Videoserver einzurichten, die Hardware-Voraussetzungen für den mehrkanaligen HD-Ingest zu schaffen und das System für die Außenübertragung zu testen und zu modifizieren. Dabei entstand ein einsatzfähiger Prototyp, der bis auf die SDI-I/O-Karten auf Standard-IT und Open-Source-Software basiert.

In diesem Kapitel soll die Arbeit am Videoserver und die technischen Voraussetzungen für den mehrkanaligen Ingest dargelegt werden. Dazu gehe ich auf Linux-Besonderheiten, die Anforderungen an die Hardware, die Vernetzung und zusätzliche Server-Software ein.

4.1 Das Betriebssystem Debian als Plattform

Um die IngeX-Software zu installieren, ist ein Linux-Betriebssystem erforderlich. Die Gründe liegen darin, dass die BBC Linux als Entwicklungsumgebung bevorzugt. Anspruchsvolle Software funktioniert dort effizienter als auf einer Windows-Plattform.¹³⁵ Betriebssysteme sollen nicht mehr nur die traditionellen Nischen füllen, sondern gezielt auf Anwendungen zugeschnitten sein. Ein Linux-System wird dafür durch seinen modularen Aufbau als geeigneter angesehen. Da das System völlig offen ist, lässt sich gezielt die Leistungsfähigkeit erhöhen. Als Anwender kann man zielgerichtet auf die nötigen Systemressourcen zugreifen und sie effizienter verwalten. Erst dadurch kann IngeX sein Potential ausschöpfen.

Linux wird als freie Software (Open Source) entwickelt. Daher existieren viele verschiedene Distributionen des Betriebssystems, die sich teilweise deutlich voneinander unterscheiden, aber grundsätzlich nach dem selben Prinzip aufgebaut sind. IngeX wurde für *openSUSE*-Linux entwickelt. Dadurch sind Anpassungen nötig, wenn die Software nicht auf openSUSE genutzt wird.

Da im grafischen und administrativen Aufbau von openSUSE immer noch zu viel Ballast vorhanden ist, sollte IngeX nun auf der Distribution Debian installiert werden. Debian bietet sich durch seinen schlanken, modularen Aufbau besonders für die Administration von Serversystemen an. Laut einer Umfrage war es 2009 das verbreitetste freie Server-Betriebssystem in deutschen Unternehmen.¹³⁶

Um den Quellcode von IngeX auf Debian zu kompilieren, mussten zahlreiche kleinere Modifikationen vorgenommen werden. Ich stellte mich der Herausforderung, um den Videoserver so minimalistisch und effizient wie möglich zu bauen. Dabei musste ich jedoch feststellen, dass die Anpassung einen beträchtlichen Zeitaufwand erfordert.

4.2 Anforderungen an die Hardware

So viel Wert auf ein möglichst schlankes Betriebssystem zu legen, hat seine Gründe. Einer der wichtigsten ist die reine Systemleistung. Die simultane

¹³⁵Vgl. Singleton: Linux at the BBC S. 48

¹³⁶Vgl. Diedrich: Trendstudie Open Source - Eingesetzte Produkte, unter: Heise online (abgerufen am 21.06.2011)

Aufzeichnung mehrere HD-SDI-Signale verlangt enorme Rechenkapazitäten. Der PC, in dem Fall Standard-IT, muss in der Lage sein, in Echtzeit mehrere Videoströme zu enkodieren und als MXF-Files zu wrappen. Dazu sind Multithread-Implementationen mit passender Hardware nötig, die zur parallelen Datenverarbeitung in der Lage sind.¹³⁷

4.2.1 Mehrkernprozessor und Arbeitsspeicher

Mit gewöhnlichen PC-Workstations ist es erst seit ungefähr 2006 möglich, SDI-Ströme in Online-Qualität zu enkodieren.¹³⁸ Computer mit vier Prozessorkernen sind beispielsweise in der Lage, vier Ströme in DVCPRO 50 oder JPEG 2:1 in Echtzeit zu berechnen. Währenddessen können auch Versionen in Offline-Qualität berechnet werden (z.B. JPEG 15:1)

Es kommt jedoch nicht nur auf die reine Prozessorleistung der CPU¹³⁹ an. Im ganzen System dürfen keine rechtechnischen Engpässe, Flaschenhälse genannt, auftreten. Mehr System-RAM ermöglicht beispielsweise größere Encoding-Puffer. Damit kann weicher auf CPU-Spitzen reagiert werden, ohne dass Framedropping¹⁴⁰ auftritt. Der Arbeitsspeicher sollte daher ausreichend dimensioniert sein. Die Untergrenze liegt bei 4 Gigabyte.¹⁴¹

4.2.2 Speicher

Bei mehrkanaligen Aufzeichnungen werden hohe Datenraten geschrieben. 4 Kanäle DNxHD 120 stoßen mit nominell 480 MBit/s bereits an die effektiven Grenzen von hochwertigen SATA-Festplatten. Daher müssen die Schreibgeschwindigkeiten der Festplattenspeicher eines Videoserver ausreichend dimensioniert sein.

Es ist empfehlenswert, RAID-Systeme¹⁴² zu nutzen, um in gleichem Maße Geschwindigkeit und Ausfallsicherheit zu verbessern. Ein vorgefertigtes NAS (*Network Attached Storage*) bietet beispielsweise Festplatten-RAIDs, die über Netzwerk angebunden werden können. NAS-Server sind einfach

¹³⁷Vgl. WHP133 S.10

¹³⁸Vgl. WHP155 S. 6

¹³⁹Central Processing Unit - Hauptprozessor

¹⁴⁰Framedropping: Einzelbilder können nicht schnell genug kodiert werden und gehen verloren

¹⁴¹Vgl. Ingex: Hardware Requirements, unter <http://ingex.sourceforge.net/studio/> (abgerufen am 23.06.2011)

¹⁴²Redundant Array of Independent Disks, Verbund mehrerer Festplatten

4 Aufbau und Konfiguration eines Ingest-Videoservers auf Debian-Plattform

zu verwalten und bieten die notwendige Zuverlässigkeit. Es entstehen zudem weniger Kosten als bei einem kommerziellen Schnitt-Speicher-Server.

Wenn kein extra Storage mitgeführt werden soll, lässt sich der Speicher auch direkt im Server verbauen. Ein RAID mehrerer Festplatten erfüllt hier alle Anforderungen und ist zudem kostengünstig. Normale SATA-Festplatten haben sich in großangelegten Studien¹⁴³ als ebenso zuverlässig erwiesen, wie die teureren SCSI- und Fibre-Channel-Festplattensysteme.

Für lokale Aufzeichnungen im Videoserver haben wir uns aus fünf SATA-Festplatten mit jeweils 300 GB Speicher ein softwaremäßiges RAID5 gebaut. Bei RAID5 speichert jede Platte Paritätsinformationen. Es darf maximal eine Platte ausfallen, um einen Datenverlust zu verhindern. Für rund 750 Euro lassen sich somit 1,2 TB lokale Aufzeichnungskapazität schaffen. Mehr ist nicht sinnvoll, da das Ingestsystem in den meisten Fällen mit externen Speichern verbunden ist. Trotz RAID sollten auf dem internen Speicher zwecks Datensicherheit regelmäßig Backups des Systems durchgeführt werden.

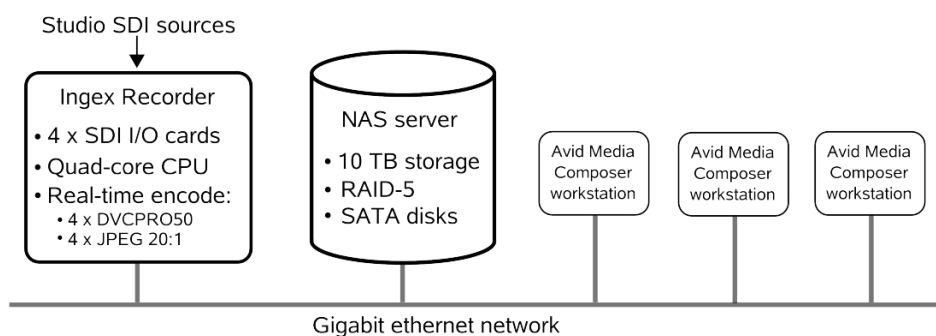


Abbildung 4.1: Beispiel eines Netzwerkes aus Recorder (Ingest), Speicher (NAS) und Clients (Avid) (Quelle: BBC White Paper 155)

4.2.3 Netzwerk

Im Produktionseinsatz ist der Ingest-Videoserver über Netzwerk an einen Mediaspeicher (NAS, Schnitt-Speicher-Server) und/oder mehrere Schnitt-Clients angebunden. Wenn das Encoding vom Server auf einen im Netzwerk befindlichen Speicher stattfindet, müssen die oben beschriebenen 480 MBit/s über Ethernet transferiert werden. Ein Gigabit-Ethernet kann nach eigenen Tests unter Linux bis zu 95% ausgelastet werden. Daher ist der reine

¹⁴³Vgl. Schroeder/Gibson: Disk failures in real world S. 3 und 14

Transfer unproblematisch.

Falls neben den Schreibzugriffen beim Enkodieren noch massive Lesezugriffe der Schnittsysteme stattfinden, kann ein Gigabit-Netzwerk an seine Grenzen stoßen. Tests der BBC haben ergeben, dass ein Gigabit-Ethernet ausreichend ist, um mit sechs Schnittsystemen unter realen Bedingungen in SD über Netzwerk zu schneiden.¹⁴⁴ Dazu ist allerdings erforderlich, dass die Netzwerkleistung auf dem jeweiligen Schnittsystem ausgereizt wird. Über CIFS-Dateitransfer (siehe Abschnitt 4.4 auf S. 64) sollten sich für Multicam-Online-Schnitt bis zu 500 Mbit/s bereitstellen lassen. Ein gewöhnliches Gigabit-Ethernet-Switch erfüllt die Anforderungen an ein solches Netzwerk.

Wenn HD-Material geschnitten werden soll, sind höhere Datenraten erforderlich. Im Abschnitt 5.5 (S. 77) sind eigene Tests beschrieben, die aufzeigen, wann eine 10-Gigabit-Schnittstelle eingesetzt werden sollte.

4.2.4 Capture-Karten

Um digitale Videosignale aufzuzeichnen und in Dateiform zu wandeln, werden spezielle Capture-Karten benötigt. Im Falle des IngeX-Videoservers sind Karten mit HD/SD-SDI-Eingängen Voraussetzung. Unter Kapitel 3 ist der Prozess des Capturings ausführlich beschrieben.

Die Software IngeX ist momentan nur mit SDI-Capture-Karten der Firma DVS kompatibel. Für die Nutzung andere Karten sind eigenständige Anpassungen am Quellcode nötig. Dadurch ergibt sich eine Abhängigkeit von proprietären Produkten, die auch die BBC als Entwickler eigentlich vermeiden wollte, in dem sie den modularen, offenen Aufbau von IngeX wählte. Für die Implementation von geeigneten Capture-Karten ist jedoch viel Entwicklungszeit nötig. Momentan arbeitet die BBC an der Unterstützung für Blackmagic Decklink Karten, um die Software einer größeren Zahl von Anwendern zugänglich machen zu können.¹⁴⁵

In der derzeitigen Version funktioniert IngeX mit Karten des Types DVS Centaurus II¹⁴⁶, SDStation OEM II und Atomix LT¹⁴⁷. Unser Videoserver arbeitet mit Centaurus II Karten, die HD-fähig sind und Multichannel-Betrieb

¹⁴⁴Vgl. WHP155 S.7

¹⁴⁵Vgl. <http://sourceforge.net/projects/ingex/forums/forum/531546/topic/4019477> (abgerufen am 23.06.2011)

¹⁴⁶Vgl. IngeX: Hardware Requirements, unter <http://ingex.sourceforge.net/studio/> (abgerufen am 23.06.2011)

¹⁴⁷Vgl. <http://sourceforge.net/projects/ingex/forums/forum/531547/topic/4539164> (abgerufen am 23.06.2011)

4 Aufbau und Konfiguration eines Ingex-Videoservers auf Debian-Plattform

unterstützen. Dadurch lassen sich mit zwei Karten bis zu vier SDI-Signale capturen.

Die ordnungsgemäße Installation von Ingex setzt ein Software Development Kit von DVS voraus. Wer Ingex trotzdem testen will, kann eine Dummy-Datei installieren, die den Capture-Vorgang der Hardware simuliert.¹⁴⁸

4.3 Eckdaten des Prototyps

Für den Aufbau des Videoservers stand mir Technik der MMG zur Verfügung. Einige Komponenten waren bereits für andere Systeme angeschafft worden, z.B. die noch unverzichtbaren DVS Capture-Karten. Ebenso die notwendige PC-Rechentechnik, darunter ein Mehrkernprozessor. Über Multithreading verteilt sich damit die Rechenlast während eines Ingests effizienter auf mehrere Prozessorkerne. Der Prototyp besitzt einen Intel Xeon Prozessor mit vier Kernen (Quad-Core) und 3,2 GHz Grundtaktfrequenz. Damit lassen sich je nach Komplexität der Encodings 2-4 HD-Aufzeichnungskanäle realisieren.

Komponente	Typ	Modell
Prozessor	Vierkernprozessor (Quad-Core)	Intel Xeon X5482 @ 3,2 GHz
Motherboard	Serverboard	SuperMicro X7DWE
Arbeitsspeicher	DDR2-SDRAM	4 GB
Speicher intern	Software-RAID	5x Western Digital WD3000, 10.000 U/min, 300 GB
Speicher extern	Hardware-RAID	EditShare, 12 TB
Capture-Karte	HD/SD-SDI In/Out	2x DVS Centaurus II
Netzwerk-Schnittstelle	10-Gigabit-Karte	MYRICOM Myri-10G Dual-Protocol NIC

Tabelle 4.1: Einige Komponenten des Ingex-Prototyps im Überblick

Für das Testsystem wurde intern eine Western Digital Festplatte mit 10.000 U/min genutzt, um lokal hohe Lese-/Schreibgeschwindigkeiten zu gewährleisten. Der interne Speicher wird in Zukunft auf ein softwaremäßiges RAID5 umgestellt, um Datensicherheit, Speichergröße und Geschwindigkeit zu verbessern. Ein SSD-Speicher¹⁴⁹ ist für die dauerhafte Anwendung in Ü-Wägen

¹⁴⁸Vgl. Ingex: Hardware Requirements, unter: <http://ingex.sourceforge.net/studio/> (abgerufen am 23.06.2011)

¹⁴⁹Solid State Drive: Speichermedium auf Halbleiterbasis

zu überlegen, da er keine mechanischen Teile besitzt und damit robuster ist. Als externer Schnitt-Speicher für geteilte Avid-Projekte steht ein *Edit-Share*¹⁵⁰ Shared Storage zur Verfügung. Für kleinere Anwendungen wird ein herkömmliches NAS eingesetzt.

Eine 10G-Ethernet-Karte ermöglicht Datenraten bis rund 10 Gbit/s in Voll-duplex und verhindert damit einen Flaschenhals durch fehlende Netzwerk-datenrate. Zusätzlich sind 1Gb-Ethernet-Schnittstellen vorhanden, die für die meisten Anwendungsfälle ausreichend sind.

Der Prototyp verfügt über ein redundantes Netzteil, um die Ausfallsicherheit zu erhöhen.

4.4 Zusätzliche Software

Neben der reinen Hardware, dem Betriebssystem und der Inget-Software sind noch zahlreiche zusätzliche Elemente nötig, um einen Videoserver einzurichten. Auf dem Debian-Linux lassen sich die erforderlichen Software-Pakete überwiegend unkompliziert mittels der Paketverwaltung *aptitude* installieren. Einige Anwendungen sind nachfolgend gelistet.

PostgreSQL - Datenbank-Server Mit PostgreSQL wird die Inget-Datenbank verwaltet, die die Konfigurationen der Weboberfläche speichert und bereitstellt. Metadaten und andere aufzeichnungsrelevante Informationen werden in die Datenbank geschrieben.¹⁵¹ PostgreSQL ist eine freie Software, mit der Datenbankmanagement betrieben werden kann. Sie nutzt die SQL-Datenbanksprache zur Abfrage und Manipulation von Daten in relationalen Datenbanken.

Apache - Webserver Der Apache-Webserver stellt die Inget-Weboberfläche zur Verfügung, die der Anwender über den Pfad `http://<IP-Adresse des Hosts>/inget` abrufen kann. Alternativ kann auch statt der IP-Adresse der Name des Hosts eingesetzt werden. Die Weboberfläche basiert auf Perl-CGI-Code. Damit lassen sich die Webseiten dynamisch erstellen und für das Monitoring des Inget-Systems (Recorder) nutzen.

¹⁵⁰ Hersteller von Komplettsystemen zur vernetzten Postproduktion, <http://www.editshare.com/>

¹⁵¹ Vgl. WHP155 S.7

CORBA - Naming Service Die umfangreiche CORBA-Bibliothek dient der Echtzeitkommunikation einzelner Softwaremodule untereinander. Sie stellt einen Naming Service bereit, durch den Clients bestimmte Objekte über Namen finden. CORBA steht für *Common Object Request Broker Architecture*. Es bietet eine Umgebung für die Vermittlung von plattformübergreifenden Diensten und Protokollen. Im Falle von Ingex wird es genutzt, um die Kommunikation zwischen den Recordern und der Ingex-GUI zu regeln. Dadurch können von einer grafischen Oberfläche aus mehrere Recorder auf verschiedenen PCs direkt angesteuert werden.

FUSE - Virtuelles Dateisystem Das *File System in Userspace* (FUSE) bietet eine Lösung für Linux-Systeme, um fremde Dateisysteme zu nutzen. Es konfiguriert ein virtuelles Dateisystem, durch das externe Dateien vom Mountpunkt aus wie Dateien des eigenen Dateisystems aussehen und performanter Zugriff auf sie möglich ist. Mit FUSE lassen sich beispielsweise NTFS-formatierte, externe Festplatten problemlos auf Debian einbinden.

AFP, SMB und FTP - Netzwerk- und Kommunikationsprotokolle Um während eines Ingest Dateien direkt über Netzwerk auf ein Storage oder einen Schnitt-Client zu enkodieren, ist Dateizugriff auf fremde Rechner nötig. Das Einbinden entfernter Dateisysteme geschieht dabei über bestimmte Netzprotokolle. Aktuelle Windows- und Macintosh-Computer unterstützen standardmäßig das Protokoll SMB (*Server Message Block*). Es implementiert das darauf aufbauende Netzwerkdateisystem CIFS (*Common Internet File System*) und bietet die entsprechenden Dateifreigaben.

Auf älteren Macintosh-Systemen (bis Mac OS 9.2.2) ist der Dateizugriff über Netzwerk nur mit dem *Apple Filing Protocol* (AFP) möglich. Die Schnitt-Flight-Cases der MMG laufen auf Mac OS 9, daher musste AFP implementiert werden. Linux arbeitet standardmäßig nicht mit AFP. Es existieren jedoch OSS-Projekte, die das Netzwerkprotokoll einbinden. Der Videoserver kann mit *Netatalk*¹⁵² als AFP-Server und mit *afpfs-ng*¹⁵³ als AFP-Client eingesetzt werden.

AFP- und SMB-Server wurden installiert, damit Schnittsysteme für spezielle Anwendungsfälle direkt auf dem lokalen Speicher des Ingest-Systems arbeiten können. Der Austausch zwischen Linux, Windows und Mac-Platt-

¹⁵²<http://netatalk.sourceforge.net/>

¹⁵³<http://sourceforge.net/projects/afpfs-ng/>

4 Aufbau und Konfiguration eines Ingex-Videoservers auf Debian-Plattform

formen über SMB/CIFS erschien uns nach ausgiebigen Tests am Praktikabelsten.

Für reine Datentransfers ist das *File Transfer Protocol* (FTP) im Einsatz, das über diverse verfügbare Client- und Server-Pakete auf Debian implementiert werden kann. Die Pakete *vsftpd* (Server) und *ftp* (Client) erfüllen diese Anforderung.

SSH - Einfache Steuerung über Netzwerk Um den Videoserver mobil zu steuern, ist der Zugriff per SSH (*Secure Shell*) möglich. Auf dem Server müssen dazu lediglich das Paket *OpenSSH* installiert und einige Konfigurationen vorgenommen werden. Der Dienst stellt dann eine Eingabekonzole des Servers zur Verfügung. Über ein *X-Window-System*¹⁵⁴ lässt sich zusätzlich eine GUI öffnen für besondere Anwendungen.¹⁵⁵

Der Vorteil von SSH ist, dass alle Zugriffe bequem von einem anderen im Netzwerk befindlichen Computer getätigt werden können. Die komplette Steuerung und Konfiguration des Ingest-Systems ist so möglich. Grafische Anwendungen und die Kommandozeile wurden bei unseren Tests auf Windows-Systemen mit den Tools *PuTTY*¹⁵⁶ und *Xming*¹⁵⁷ aufgerufen. PuTTY ist ein textorientierter SSH-Client für Windows, *Xming* ein X-Server zur grafischen Anzeige von Anwendungen über SSH. Ein X-Server läuft auf dem lokalen Arbeitsplatzrechner¹⁵⁸ und bietet die grafische Schnittstelle für Server-Anwendungen, in unserem Fall z.B. die Ingexgui, mit der das Recording gesteuert wird.

¹⁵⁴Sammlung von Protokollen, Programmen und Standards zur Anzeige einer grafischen Benutzeroberfläche

¹⁵⁵Vgl. Amberg: Linux-Server mit Debian GNU/Linux S. 324

¹⁵⁶<http://www.putty.org/>

¹⁵⁷<http://sourceforge.net/projects/xming/>

¹⁵⁸Vgl. Amberg S. 325

5 Einsatz in der Außenübertragung

Zahlreiche Veränderungen an der Ingex-Software sind nötig, um ein in der Fernseh-Außenübertragung (AÜ) taugliches und leistungsfähiges System zu entwickeln. Die Anforderungen von Außenproduktionen sind teils sehr speziell. Sie erfordern einen flexibel konfigurierbaren, automatisierten, einfach zu bedienenden und sehr schnellen Aufzeichnungsserver. Schnelligkeit vor allem im Bezug auf die Bereitstellung des Material für den non-linearen Schnitt. In diesem Kapitel soll dargelegt werden, welche Anpassung an unserem Ingex-Prototypen vorgenommen worden, um das zu realisieren.

Dazu gehe ich auf einige Grundgedanken für den Einsatz in der AÜ ein und beschreibe, wie man mittels Edit-While-Capture und Automatisierung die Produktion mit Ingex effektivieren kann. In einem weiteren Punkt werden in Tests die rechnerischen Grenzen des Systems ermittelt.

5.1 Allgemeine Grundgedanken

Von Anfang an war das Ziel meiner Arbeit, einen lauf- und einsatzfähigen Ingex-Prototypen für die Fernseh-Außenübertragung zu bauen. Ingex wurde ursprünglich als Aufzeichnungslösung in einer Studioumgebung entwickelt und hat eine darauf spezialisierte Funktionalität. Die Software ist jedoch so flexibel aufgebaut, dass sich zahlreiche Möglichkeiten bieten, sie auf eigene Bedürfnisse anzupassen.

In der AÜ bieten sich dauernd wechselnde Produktionsszenarien und -bedingungen. Ein Aufzeichnungsserver muss sich auf diese möglichst einfach zuschneiden lassen. Das erforderte eine gute Bedienbarkeit und ein hohes Maß an Flexibilität und Robustheit. Variable Aufzeichnungscodecs und einfache Bedienoberflächen sind Grundvoraussetzungen. Ein modularer Aufbau ermöglicht beispielsweise, die Anzahl der Kanäle zu erweitern und be-

stimmte Teile der Software zu aktivieren/deaktivieren. Um das alles zu realisieren, müssen Kompromisse eingegangen werden. Der Ingex-Videoserver wurde von uns so konzipiert, dass er im Idealfall komplett vorkonfiguriert wird, einfach in jedes System integrierbar ist und sofort einzusetzen ist, auch für Nutzer, die keine speziellen Kenntnisse besitzen.

Der spezialisierte Ingex-Prototyp soll als mobiler Videoserver in der AÜ Einsatz finden, um SDI-Signale in einen dateibasierten Datenstrom umzusetzen. Wichtig ist dabei der unkomplizierte Einbau in einen bestehenden Übertragungswagen. Mittels gängiger Verkabelung (HD/SD-SDI eingangsseitig, RJ45 ausgangsseitig), 19-Zoll-Gehäuse und vorkonfigurierter Software ist das Gerät einfach in eine Ü-Wagen-Infrastruktur zu integrieren. Die Steuerung kann über Netzwerk (SSH) oder direkt erfolgen.

Programmbilder oder einzelne Quellen wie z.B. Kamerasignale lassen sich unkompliziert für Schnittsysteme aufzeichnen und zur Bearbeitung im Netzwerk bereitstellen. Damit bietet Ingex eine Alternative zu kostenintensiven, proprietären Systemen bzw. kann auch ergänzend genutzt werden, um die Zahl der Aufzeichnungskanäle zu erhöhen. Es entkoppelt den Ingest vom Schnittsystem und von Herstellerabhängigkeit.

5.2 Edit-While-Capture

Im Gegensatz zur Studioproduktion ist in der AÜ eine zeitnahe Postproduktion von Live-Events meist unumgänglich. Material wird aufgezeichnet und muss während oder kurz nach der Aufzeichnung fertig bearbeitet sein. Im Studiobereich war der Zeitgewinn einer dateibasierten Aufzeichnung ausreichend, um sich mühsames Einspielen von Bändern zu sparen. Daher ist Ingex so konzipiert, dass das Material erst nach Aufzeichnungsende zur Verfügung steht und bearbeitet werden kann.

In Außenproduktionen muss Material schneller verfügbar sein. Wenn z.B. ein Sportwettkampf aufgezeichnet wird und am Ende eine Zusammenfassung der Höhepunkte ausgestrahlt werden soll, steht wenig Zeit für die Bearbeitung zur Verfügung. Ein Cutter kann häufig nicht erst am Ende des Wettkampfes mit dem Schnitt beginnen. Er muss während der Aufzeichnung schneiden können, damit eine Zusammenfassung am Ende sofort bereit steht.

Daher war mein Hauptziel, jenen Schnitt während der Aufzeichnung (Engl.:

Edit-While-Capture¹⁵⁹, kurz EWC) mit Ingeg zu ermöglichen. Wenn EWC mit non-linearen Schnittprogrammen wie dem Avid Media Composer oder Final Cut Pro geschehen soll, muss zum einen in nativen Bearbeitungsformaten (wie MXF OP-Atom oder QuickTime) aufgezeichnet werden, zum anderen müssen „anwachsende“ Dateien erzeugt werden, auf die Schnitt-Clients bereits während des Schreibprozesses zugreifen können.

Nach langen Tests und Versuchsreihen konnte ich einen Edit-While-Capture-Workflow auf Basis des MXF OP-Atom für den Avid Media Composer einrichten. Als Grundlage für meine Versuche diente der Highlighting- und Slowmotion-Videoserver *EVS XT2*. Dort ist ein solcher Workflow mit den Dateiformaten AAF und MXF möglich. Ich analysierte die Funktionsweise der EVS sehr genau, um sie für Ingeg zu adaptieren. Dabei nutzte ich die Möglichkeiten, die die offenen Formate AAF und MXF von Grund auf bieten.

5.2.1 Analyse eines Vergleichssystems

Die EVS XT2 ist ein verbreitetes Mehrkanalaufzeichnungsgerät der Firma EVS Broadcast Equipment, das in Endlosschleife SDI-Signale mitzeichnet, die mit Controllern geschnitten und als Clips abgelegt werden können. Die EVS-Tools XTAccess und IPDirector ermöglichen den Zugriff auf aufgezeichnetes Material per Netzwerk und stellen es für Schnittsysteme bereit. Dabei können die Daten in die jeweiligen nativen Verarbeitungsformate gewrappt werden, so auch MXF OP-Atom für Avid-Clients. Der IPDirector erzeugt zusätzlich AAF-Dateien, die eine Bearbeitung von Material während des Transfers in eine Avid-Umgebung zulassen.

Die EVS XT2 zeigt, dass lediglich auf Basis der offenen, standardisierten Formate MXF und AAF ein EWC-Workflow erreicht wird. Daher diente es mir als Modellsystem, das ausgiebig analysiert wurde. Zunächst musste ich herausfinden, ob bestimmte Daten in die MXF- oder AAF-Dateien geschrieben werden, die einem Avid-Client signalisieren, dass ein EWC stattfinden soll. Von großer Relevanz ist zudem, zu welchem Zeitpunkt während einer EWC-Aufzeichnung MXF und AAF bearbeitet werden und wie sie strukturell aufgebaut sind. Im zweiten, viel komplizierteren Schritt, müssen die Erkenntnisse aus dem EVS-System für Ingeg adaptiert werden.

¹⁵⁹ andere Bezeichnungen: Edit-While-Ingest, Edit-While-Recording

5.2.1.1 Auslesen der MXF- und AAF-Container

Ein exaktes Verständnis des Workflows war nötig, um EWC für IngeX zu realisieren. Es wurde ein Testsystem mit einer EVS XT2 aufgebaut, bei dem alle erzeugten Dateien während und nach Abschluss des Prozesses analysiert wurden. Zur Auswertung mussten die von der EVS geschriebenen MXF/AAF-Files ausgelesen und mit den IngeX-Files verglichen werden. Auszüge einiger Dump-Dateien sind im Anhang zu finden.

Da die Daten KLV-codiert sind, müssen sie erst korrekt in Textform übersetzt werden, um sie auswerten zu können. Dieser Prozess nennt sich „Dumping“, was computertechnisch soviel heißt wie den „Speicherinhalt anzeigen“. Sowohl die „libMXF“-Bibliothek von IngeX (*MXFDump*), als auch das AAF Toolkit (*InfoDumper*) stellen Werkzeuge bereit, die MXF/AAF-Dateien auslesen können. Zusätzlich habe ich die freie MXF-Bibliothek „MXF-Lib“¹⁶⁰ genutzt, die zahlreiche Tools zum Bearbeiten von MXF-Dateien bietet. Mit ihr können Dateien ausgelesen (*MXF-Dump*), Essenz extrahiert (*MXF-Split*), Essenz in MXF-Container gepackt (*MXF-Wrap*) oder Dateistrukturen grafisch dargestellt werden (*MXF-2dot*).

5.2.1.2 Auswertung der Dumps

Regelmäßige Dumps während der Aufzeichnung gaben mir einen Aufschluss darüber, wann welche Partitionen im MXF geschrieben werden. Die aus Tabelle 5.1 folgenden Anpassungen an IngeX sind unter 5.2.2 beschrieben.

Objekt	anwachsendes EVS-MXF	anwachsendes IngeX-MXF	abgeschlossenes MXF
Header	vollständig, geschlossen	unvollständig, geschlossen	vollständig, geschlossen
Body	unvollständig	unvollständig	vollständig, geschlossen
Footer	unvollständig, vorhanden	nicht vorhanden	vollständig
Index-Tabelle	nicht vorhanden	nicht vorhanden	vorhanden
Cliplänge (Duration)	unbekannt	unbekannt	fester Wert
OperationalPattern	MXF Specialized OP Atom	MXF Specialized OP Atom	MXF Specialized OP Atom

Tabelle 5.1: MXF OP-Atom während der Aufzeichnung mit EVS und IngeX, verglichen mit einem abgeschlossenen File (Quelle: eigene Darstellung)

¹⁶⁰<http://freemxf.org/>

5 Einsatz in der Außenübertragung

Die Analyse der AAF-Dateien ergab zudem:

- EWC erfordert eine festgelegte Cliplänge in der AAF-Datei. Die „length“ von Video und Audio Tracks muss einen festen Wert größer 0 Frames haben. Für Clips unbekannter Länge wird im Normalfall -1 angegeben.
- Es existieren Metadaten, die einem Avid-Client signalisieren, dass Edit-While-Capture stattfindet. Sie sind in einem EWC-Bündel und einem OMM-Bündel zusammengeschlossen.
- Eine gesonderte Attributliste im AAF, die *MobAttributeList*, übergibt diese Metadaten an den Avid.

Das EWC-Bündel enthält die erwartete Länge des Clips (*EWC EXPECTED DUR*), ein im Voraus festgelegter Wert. Er übergibt dem Avid ein temporäres Ende der Datei. *EWC KNOWN MEDIA DUR* ist der Wert, der die tatsächliche Cliplänge beschreibt, die verfügbar ist. Er wird regelmäßig aktualisiert. Der dritte Wert *EWC MOD TIME* ist vermutlich ein eindeutiger Zeitwert, der einem EWC-Paket zugewiesen wird und der Identifikation dient. Dazu ließen sich allerdings keine genauen Angaben finden.

Das OMM-Bündel signalisiert mit dem Wert *OMM MOB COMPLETE* vor allem, ob ein Edit-While-Capture abgeschlossen ist (Wert 1) oder noch läuft (Wert 0). Es handelt sich um ein Attribut des *Open Media Management* (OMM). OMM-Attribute können z.B. zu einer *Avid Unity* Umgebung gehören.¹⁶¹ Sie dienen dem Austausch von Schnitt-Daten und Essenz über Netzwerke. Über das Setzen dieser Metadaten in AAFs können mit Inqex die EWC-Workflows für Avid erreicht werden.

Das Open Media Management ist von Avid entwickelt worden, um fremden Systemen eine Schnittstelle zu Avid-Clients zu bieten.¹⁶² Besonders sogenannte Asset Management Systeme von Drittherstellern sollen eingebunden werden. Asset Management Systeme sind Computerserver, die mediale Inhalte wie Video-, Audio-, Grafik- und Textdateien inklusive Metadaten speichern, verwalten und archivieren.¹⁶³ Sie sind beispielsweise mit Produktions- oder Sendeservern verbunden, die auf das Material zugreifen.

¹⁶¹ Vgl. Avid: Unity MediaManager S. 13

¹⁶² Ebd. S. 13

¹⁶³ Vgl. BET Fachwörterbuch: Asset Management System, unter: <http://www.bet.de/Lexikon/Begriffe/assetmanagementsystem.htm> (abgerufen am 03.07.2011)

5 Einsatz in der Außenübertragung

```
774 MobAttributeList: Strong Object Reference Array [4] of TaggedValue
775   kAAFTypeID_TaggedValueStrongReferenceVector [0]:
776
777     Object of Class: TaggedValue
778     Name: "_EWC_BUNDLE"
779     Value: (indirect type: aafString) "__AttributeList"
780     TaggedValueAttributeList: Strong Object Reference Array [3] of TaggedValue
781     kAAFTypeID_TaggedValueStrongReferenceVector [0]:
782
783       Object of Class: TaggedValue
784       Name: "_EWC_EXPECTED_DUR"
785       Value: (indirect type: aafInt32) 450000
786     kAAFTypeID_TaggedValueStrongReferenceVector [1]:
787
788       Object of Class: TaggedValue
789       Name: "_EWC_KNOWN_MEDIA_DUR"
790       Value: (indirect type: aafInt32) 27000
791     kAAFTypeID_TaggedValueStrongReferenceVector [2]:
792
793       Object of Class: TaggedValue
794       Name: "_EWC_MOD_TIME"
795       Value: (indirect type: aafInt32) 1302610796
796
797   kAAFTypeID_TaggedValueStrongReferenceVector [1]:
798
799     Object of Class: TaggedValue
800     Name: "_OMM_BUNDLE"
801     Value: (indirect type: aafString) "__AttributeList"
802     TaggedValueAttributeList: Strong Object Reference Array [2] of TaggedValue
803     kAAFTypeID_TaggedValueStrongReferenceVector [0]:
804
805       Object of Class: TaggedValue
806       Name: "_OMM_MOB_COMPLETE"
807       Value: (indirect type: aafInt32) 0
808     kAAFTypeID_TaggedValueStrongReferenceVector [1]:
809
810       Object of Class: TaggedValue
```

Abbildung 5.1: InfoDumper-Auszug aus einem AAF - Die EWC-Attribute liegen als Metadaten vor (Quelle: eigene Darstellung)

Abbildung 5.1 zeigt die im AAF eingebetteten Metadaten. Das IngeX-Modul Createaaf wurde dank der umfangreichen Hilfe von BBC-Entwickler Philip de Nier von mir so abgeändert, dass die notwendigen EWC/OMM-Attribute in die AAF-Dateien geschrieben werden können. Damit kann der EWC-Workflow mittels AAF eins zu eins für IngeX adaptiert werden. Die zusätzlichen Änderungen sind folgend beschrieben.

5.2.2 Zugriff auf anwachsende MXF-Dateien mit Avid

Um einem Avid-Editor sofortigen Zugriff auf Material zu ermöglichen, das gerade gecaptured wird, sind weitere Anpassungen an IngeX nötig. Drei Bedingungen mussten geschaffen sein, damit der Avid anwachsende MXF-Dateien nicht sofort als korrupt erkennt oder sie ignoriert: ¹⁶⁴

- Ein sauber kodierte Operational Pattern MXF OP-Atom muss vorliegen.
- MXF-Dateien müssen direkt ins Avid-Watchfolder geschrieben werden ohne Creating-Unterordner.
- Während des Encodings muss ein temporärer Footer und ein abgeschlossener Header ins MXF geschrieben werden, damit ein Avid-Client darauf zugreifen kann.

Zunächst müssen die anwachsenden MXF-Files direkt in ein Avid-Watchfolder¹⁶⁵ gewrappt werden. Der Zielpfad für das IngeX-Encoding wird dazu einfach in der Weboberfläche festgelegt.

IngeX nutzt aber standardmäßig (wie auch Avid) den temporären „Creating“-Unterordner für einlaufendes Material. Erst wenn die Files abgeschlossen sind, werden sie in den Avid-Watchfolder kopiert und damit freigegeben. Um sofortigen Zugriff für den Editor zu erreichen, muss der Zielpfad für das Encoding im IngeX-Quellcode geändert werden. Dabei muss bedacht werden, dass das Transfermanager-Modul von IngeX daraufhin nicht mehr korrekt funktioniert, da dieser sonst nur partiell geschriebene Dateien kopiert.

Leider kann Avid standardmäßig nicht mit noch unvollständigen IngeX-MXF-Files umgehen. Daher muss normalerweise immer das Ende einer

¹⁶⁴ getestet für Avid Media Composer 3.5 und 4.0

¹⁶⁵ Pfadstruktur: <Laufwerk>/Avid MediaFiles/MXF/<Ordner mit Laufnummer>

Aufzeichnung abgewartet werden. Bei der Analyse von anwachsenden MXF-Dateien, die der Avid Media Composer verarbeiten kann, stellte sich heraus, dass der Avid immer einen abgeschlossenen Header und einen temporären Footer verlangt. Ingex schreibt allerdings in seine anwachsenden MXF-Files nur einen temporären Header und gar keinen Footer.

Mit einigen Veränderungen im Ingex-Quellcode, der für das MXF-Wrapping zuständig ist, konnten jedoch erfolgreich anwachsende MXF-Files erzeugt werden, die der Avid interpretieren kann. Dazu wurde das Modul Writeavid-mxf (Abschnitt 3.4.3 auf S. 43) der MXF-Bibliothek libMXF angepasst. Der Header ist nun ClosedComplete statt ClosedIncomplete während des MXF-Wrappings, der Footer wird temporär als ClosedIncomplete geschrieben.

Bei den Änderungen am Quellcode half mir erneut der BBC-Mitarbeiter Philip de Nier, der am Ingex-Projekt arbeitet. Nur in Zusammenarbeit mit ihm konnte ich die Software nach meinen Vorstellungen anpassen. Er verhalf mir mit seinen Kenntnissen vom programmiererischen Konstrukt, meine Erweiterungen zu implementieren.

5.2.3 Edit-While-Capture mit AAF-Dateien realisieren

Mit den bisherigen Änderungen kann der Avid zwar prinzipiell die MXF-Dateien lesen, ihm fehlt jedoch der Hinweis darauf, dass es sich um anwachsende Files handelt. Avid sieht die Bearbeitung von anwachsenden EWC-Clips grundsätzlich vor. Er kann einen sogenannten *In-Masterclip* für noch nicht abgeschlossene Aufnahmen erstellen. Das ist ein Masterclip, der sozusagen „in Arbeit“ ist. Er hat eine festgelegte Länge und einen besonderen Icon.



In-progress master clip

Abbildung 5.2: Avid stellt anwachsende MXF-Dateien mit In-progress Clips dar (Quelle: Avid)

An dem Punkt werden Metadaten bedeutsam. Bestimmte in AAF und MXF abgelegte Metadaten indizieren dem Avid, dass solch ein Clip vorliegt (siehe Abschnitt 5.2.1 auf S. 68). Die AAF-Dateien referenzieren anwachsende MXF-Files und übergeben gleichzeitig die Avid-spezifischen EWC-

5 Einsatz in der Außenübertragung

Metadaten. Sie lassen sich einfach per Drag&Drop in die Bin eines Avid-Projektes ziehen. Ein In-progress Masterclip wird daraufhin automatisch angelegt. Die Abspielen und die Bearbeitung von anwachsenden MXF-Dateien ist sofort möglich.

Unvollständiges, noch nicht aufgezeichnetes Material wird durch ein Standbild mit dem Hinweis „Capture in Progress“ gekennzeichnet (Abbildung 5.3), während die schon ins MXF geschriebenen Daten am Anfang des Clips voll verfügbar sind. Der In-progress Masterclip aktualisiert sich automatisch, so dass das Material im Clipfenster anwächst. Avid nennt dieses Feature auch „Frame Chase Editing“.¹⁶⁶

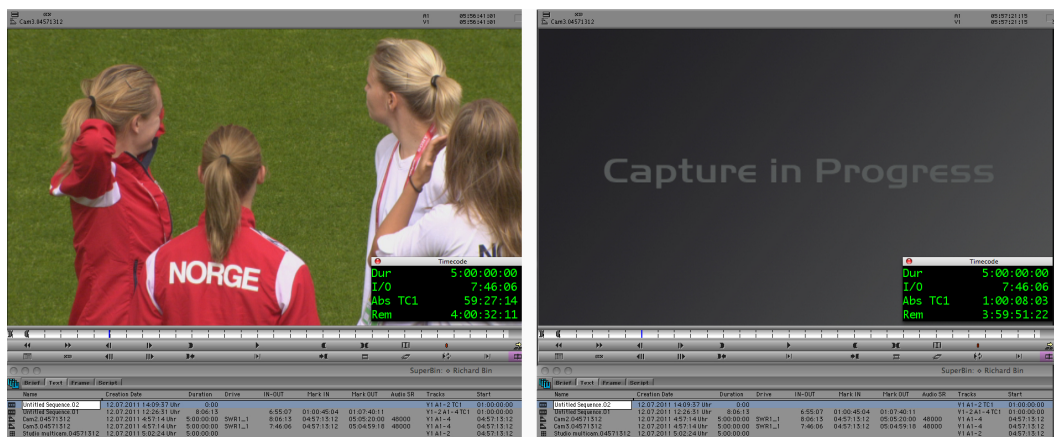


Abbildung 5.3: Capture in Progress im Avid - Bereits verfügbares Material wird angezeigt (links), während noch der noch nicht aufgezeichnete Teil (rechts) „in Progress“ ist (Quelle: Screenshot auf Avid-PC)

Frame Chase Editing ist eigentlich für eine *Avid Interplay* Umgebung entwickelt worden.¹⁶⁷ Es erlaubt dem Editor, Material während des Capture-Vorgangs abzuspielen und zu schneiden. Der Schnitt einer Sequenz kann fertiggestellt werden noch bevor die Aufzeichnung zu Ende ist. Die Beschränkung der In-progress Clips sind:¹⁶⁸

- **Schnitt:** Nur bereits aufgezeichnete Teile können in eine Sequenz geschnitten werden.
- **Subclips:** Subclips müssen eine In- und eine Out-Marke in dem Bereich haben, der verfügbar ist.

¹⁶⁶Vgl. Crooks: Interfacing K2 to Avid Editing Systems S. 3

¹⁶⁷Avid: Interplay Best Practices S. 133

¹⁶⁸Ebd. S. 139

- **Trimmen:** Rechtsseitiges Trimmen ist ebenfalls vom verfügbaren Material begrenzt.
- **Duplizieren:** In-progress Clips können nicht dupliziert werden, da sonst die Referenz zum anwachsenden MXF verloren geht.
- **Gruppierung:** Gruppierte Clips sind nicht möglich, da die Länge der Clips dazu bekannt sein muss.
- **Umwandlungen:** Konsolidieren, Transcodieren und Audiokonversion ist erst möglich, wenn die Aufzeichnung abgeschlossen ist. Allerdings können Subclips und Sequenzen umgewandelt werden, wenn sie nur auf bereits verfügbares Material verweisen
- **Variable Bitraten:** Frame Chase Editing funktioniert für Videokompressionen mit variabler Bitrate (variable Framegröße) nur bedingt. Avid gibt an, dass nur Low-Res long-GOP MPEG-2 unterstützt wird.

5.2.4 Anpassungen am Ingeg-Workflow

Die AAF-Dateien müssen die korrekten Attribute beinhalten für einen fehlerfreien EWC-Workflow. Ingeg erstellt ein AAF aus seiner Datenbank. Die gewonnen Erkenntnisse aus den Tests und Vergleichen ergaben daher folgende Arbeitsweise:

1. Beginn einer Aufzeichnung.
2. Anwachsende MXF-Dateien mit temporärem Footer und vollständigem Header werden ins Avid-Watchfolder geschrieben.
3. Einlesen der MXF-Dateien in die Datenbank mittels des Ingeg-Tools *import_mxf_info*.
4. Bearbeiten der Cliplänge in Datenbank: Length wird von -1 (unbekannt) auf 450.000 (5h) gesetzt.
5. Auslesen des MXF-Erstellungsdatum (Creation Date) aus der Datenbank, um AAF korrekt zu referenzieren.
6. Erstellen des AAF mittels Creation Date durch das Ingeg-Modul *Create_aaf*. Die EWC/OMM-Attribute werden in das AAF geschrieben.

7. Drag&Drop des AAF in den Avid-Client. Die anwachsenden MXFs werden referenziert. Edit-While-Capture ist möglich.

5.3 Automatisierung

Um den Ingest-Server schnell und unkompliziert nutzen zu können, sind automatisierte Workflows empfehlenswert. Die Zielsysteme, wie z.B. ein Schnitt-Speicher oder ein Avid-Client, müssen direkt angesteuert werden. Dazu ist notwendig, dass der Videoserver die Zielsysteme und deren Struktur automatisch erkennt. Er muss regelmäßig den Status der im Netzwerk befindlichen Geräte abfragen und sie identifizieren können. Schreibrechte und Freigaben des Zielverzeichnis werden kontrolliert.

Die Bedienung eines Ingests läuft über eine Weboberfläche und die Ingestgui. Mit dem Start eines Recordings werden verschiedene Prozesse angestoßen, die die Metadaten für das Zielsystem bereitstellen, Metadaten in die Datenbank einlesen und AAF-Dateien erstellen. Alle Prozesse sollen automatisch eingeleitet werden, um sowohl Zeit zu sparen, als auch die Bedienung möglichst einfach zu halten.

Automatisierte Abläufe wurden im Prototyp größtenteils über *Shellskripting* umgesetzt. Shellskripte sind kleine Programme, die sich für unkomplizierte Routineaufgaben einsetzen lassen. Sie können direkt in der Linux-Umgebung geschrieben werden und sind äußerst flexibel. Ein Shellskript besteht aus Benutzerabfragen, Bildschirmausgaben, bedingten Verzweigungen, Schleifen und vor allem aus normalen Linux-Kommandos.¹⁶⁹ Die Skripte müssen nicht kompiliert werden, sondern werden von einem externen Interpreter Stück für Stück abgearbeitet. Daher liegen sie immer in Textform vor, deren Bearbeitung mit einem simplen Editor möglich ist.

5.4 Bedienung und Konfiguration

Die Automatisierung soll gleichzeitig die Bedienbarkeit verbessern. Ein mit dem System nicht vertrauter Cutter muss den Ingest intuitiv steuern können. Dazu sind simple, selbsterklärende Oberflächen notwendig. Hinter ihnen verbergen sich zwangsweise komplexe Prozesse. Die Oberflächen von

¹⁶⁹Vgl. Amberg S. 268

Ingex sollen teilweise genutzt werden, allerdings ist eine eigene grafische Weboberfläche angestrebt, über die alle Vorgänge gesteuert werden können. Sie ist zum Zeitpunkt der Bachelorarbeit noch nicht umgesetzt.

Eine simple Bedienbarkeit erfordert, dass der Administrator des Videoservers das System vor einem Produktionseinsatz komplett vorkonfiguriert. Dazu steht ihm die Ingex-Weboberfläche (siehe Abschnitt 3.4.7 auf S. 49) und zahlreiche Skripte zur Verfügung. Veränderungen an den Recorder-Einstellungen können auch direkt in der Datenbank vorgenommen werden. Die Skripte lassen sich für die jeweiligen Anwendungsfälle anpassen und miteinander kombinieren. Angestrebt ist, dass anliegende Signale automatisch identifiziert werden und Skripte/Prozesse sich automatisch darauf anpassen.

Das ist die Voraussetzung für eine einfache, unkomplizierte Implementierung des Videoservers in einen Übertragungswagen. Da das System nur als Schnittstelle zwischen SDI-Geräten und dateibasiertem Schnitt dient, ist es idealerweise entsprechend flexibel und passt sich intelligent auf die angeschlossenen Systeme an.

5.5 Praxistests zu Belastbarkeit und Grenzen des Systems

Der Ingex-Video-server und ein angehängter Schnitt-Speicher müssen so dimensioniert sein, dass auf mehreren Kanälen HD-SDI-Signale gecaptured, enkodiert, gewrappt und in Echtzeit für Schnittsysteme bereitgestellt werden. Dazu war festzustellen, welchen rechnerischen Belastungen die Infrastruktur standhalten muss und wo die Grenzen des Standard-IT-Systems sind. Insbesondere die CPU-Last stellte sich in vorherigen Tests und nach Ausschluss anderer potentieller Flaschenhälse als entscheidend für die maximale Zahl der zu kodierenden AV-Ströme heraus.

Interessant schien zudem, welchen Belastungen das Netzwerk standhalten muss. Vor allem, wenn auf externe, über CIFS eingehängte Speicher aufgezeichnet wird oder multiple Zugriffe durch mehrere Avid-Clients auf anwachsendes Material stattfinden.

5 Einsatz in der Außenübertragung

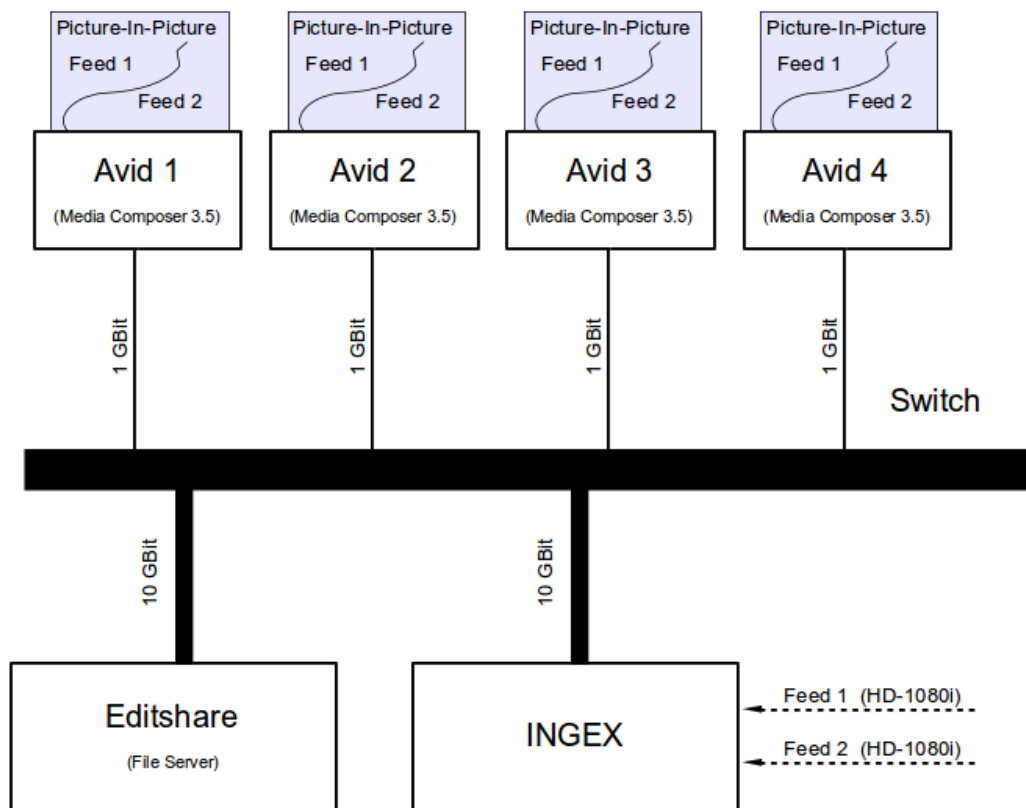


Abbildung 5.4: Schematische Darstellung der Versuchsanordnung für 2 HD-Kanäle (Quelle: eigene Darstellung)

Über einen Versuchsaufbau (Abbildung 5.4) protokollierte ich für bis zu 4 HD-Aufzeichnungskanäle die prozentuale Auslastung der CPU und die Datenraten im Netzwerk. Als Aufnahmeformat diente *DNxHD 120i* in einem MXF OP-Atom-Container. Das Avid-Format verbindet eine hohe Datenrate von 120 MBit/s mit hohem Encodingaufwand und ist ein für die MMG relevantes Postproduktionsformat. Die Ergebnisse der Untersuchungen sind in diesem Abschnitt dargelegt.

5.5.1 Anzahl der Aufzeichnungskanäle

Testläufe zeigten, dass unser Ingex-Prototyp mit steigender Kanalzahl an seine CPU-Leistungsgrenzen stößt. Um einen sicheren Betrieb bei Fernsehproduktionen zu gewährleisten, dürfen CPU-Spitzen nicht dazu führen, dass der Echtzeit-Enkodiervorgang unterbrochen wird. Sonst würden Frames verloren gehen. Die Aufzeichnung wäre unvollständig und würde im schlimmsten Fall komplett abbrechen.

5 Einsatz in der Außenübertragung

Die Ingeg-Module *Recorder*, *Ingexgui* und das Capturing sind die Hauptprozesse, die während einer Aufzeichnung die CPU beanspruchen. Mit dem Linux-Tool *top* habe ich ihre prozentuale CPU-Nutzung überwacht. Alle 5 Sekunden wurden die *top*-Werte in eine Textdatei geschrieben. Das folgende Diagramm zeigt, wie die Auslastung steigt, wenn die Zahl der HD-Kanäle erhöht wird.

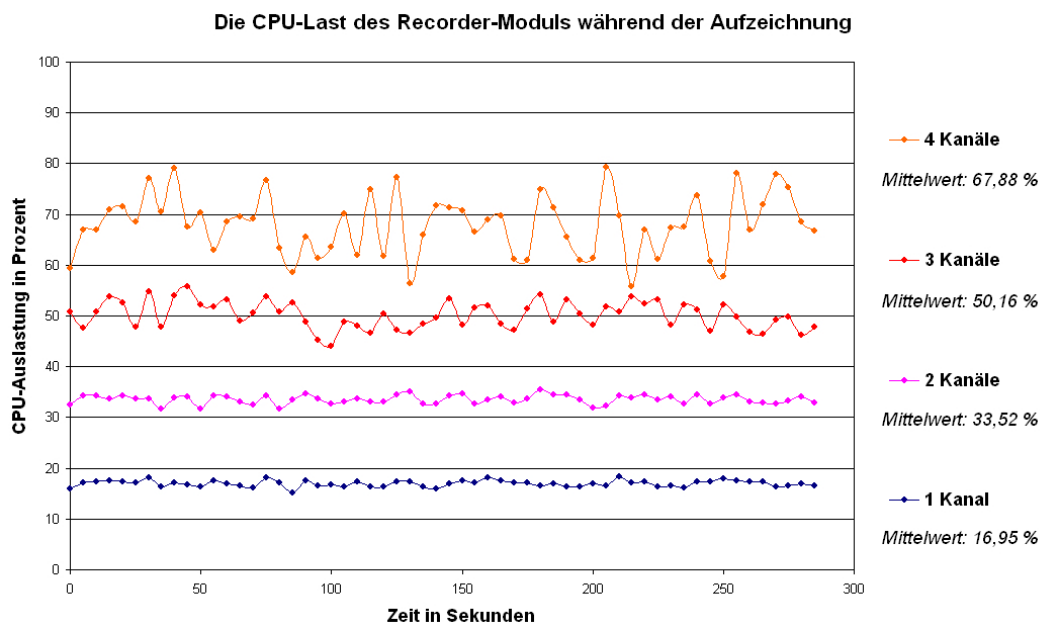


Abbildung 5.5: Die Recorderprozesse während des Ingests mit bis zu 4 HD-Kanälen (Quelle: eigene Darstellung)

Die Tests zeigen, dass die CPU-Last des Recorders (Abbildung 5.5) linear steigt mit der Anzahl der Kanäle. Der Recorder kodiert in Echtzeit die anliegenden Signale, daher benötigt er den höchsten Rechenaufwand. Die GUI und der Capture-Prozess (*dvs_sdi*) verlangen weniger Prozessorlast (siehe Tabelle 5.2), vor allem das Capturing ist jedoch bei mit steigender Kanalzahl zu beachten. Die Ergebnisse machen zudem deutlich, dass die Aufzeichnung mit 3 Kanälen problemlos möglich ist, ohne dass Frames gedroppt werden. Die Auslastung ist mit rund 50 % (Gesamt 65 %) in einem angemessenen Bereich.

Für einen vierten 4. Kanal steigt die Gesamt-CPU-Last auf durchschnittlich 84 % mit Spitzen bis zu 98 %. Daher läuft der Encoding-Puffer im RAM in unbestimmten Abständen voll und Frames gehen mit hoher Wahrscheinlichkeit verloren. Wir beschränken uns bei der MMG aus diesem Grund derzeit auf die Aufzeichnung von 3 Kanälen in HD.

Kanäle	Recorder	Capture (dvs_sdi)	Ingexgui	Gesamt-CPU-Last	CPU-Spitzen
1	17 %	4 %	3 %	24 %	26 %
2	34 %	4 %	3 %	41 %	43 %
3	50 %	13 %	2 %	65 %	70 %
4	68 %	14 %	2 %	84 %	98 %

Tabelle 5.2: Die durchschnittlichen CPU-Lasten für 1-4 Kanäle im Vergleich mit der jeweiligen Spitzenlast (Quelle: eigene Darstellung)

Eine DVS-Capture-Karte hat allerdings 2 HD-Eingänge. 1 Kanal der zweiten Karte wäre somit ungenutzt. Deshalb streben wir an, ein zukünftiges System so zu dimensionieren, dass in einem kompakten PC mit maximal 2 Höheneinheiten und einer SDI-Karte jeweils 2 HD-Kanäle aufgezeichnet werden. Das ist ein rechentechnisch sicherer Kompromiss, der genug CPU-Last für Zweit- und Dritt-Encodings einräumt. Außerdem ist ein flexibler Einsatz in Übertragungswägen so möglich.

Im Übrigen ist die Kodierung von 4 SD-Strömen ohne weiteres realisierbar, wenn die Hardware auf 2 HD-Ströme ausgelegt ist. Das zeigten ausgiebige Testreihen mit Kompressionsformaten wie *DVCPRO 50*, *IMX 50* und *MJPEG 2:1*. Die Ursache liegt in den geringeren Datenmengen des SD-SDI-Signals (270 MBit/s¹⁷⁰ gegenüber 1,485 GBit/s¹⁷¹) und dem resultierenden geringeren Kodier-Aufwand.

Zu Tabelle 5.2 ist hinzuzufügen, dass die ähnlichen CPU-Werte beim Capturing für 1/2 und 3/4 Feeds testbedingt sind, da jeweils der zweite bzw. vierte Kanal gepuffert wurde. Die sinkende CPU-Last der Ingexgui ist eventuell darauf zurückzuführen, dass Ressourcen für den Recorder-Prozess umverteilt wurden.

5.5.2 Encoding über Netzwerk

Wenn Videodaten vom Ingex-Server direkt auf externe (Schnitt-)Speicher geschrieben werden, müssen große Datenmengen im Netzwerk bewegt werden. Um festzustellen, wie dies zu dimensionieren ist, habe ich auch hier

¹⁷⁰gemäß SMPTE 259M

¹⁷¹gemäß SMPTE 292M

5 Einsatz in der Außenübertragung

mehrkanalige Aufzeichnungen im Kompressionsformat *DNxHD 120i* durchgeführt. Ein EditShare diente als externer Schnitt-Speicher.

Die Anzahl der Aufzeichnungskanäle wurde im Versuchsaufbau schrittweise gesteigert und die Auslastung der Schnittstellen protokolliert. Mir standen zwei DVS-Capture-Karten zur Verfügung. Damit ließen sich bis zu 4 Kanäle in HD aufzeichnen. Die Gesamt-Belastung an der Netzwerk-Schnittstelle (empfangene und gesendete Bytes) wurde mit Hilfe des Werkzeugs *bwm-ng*¹⁷² gemessen. Die Datenrate wurde über ein Skript automatisch alle 5 Sekunden in eine Textdatei ausgelesen. Das folgende Diagramm zeigt die ermittelten Werte.

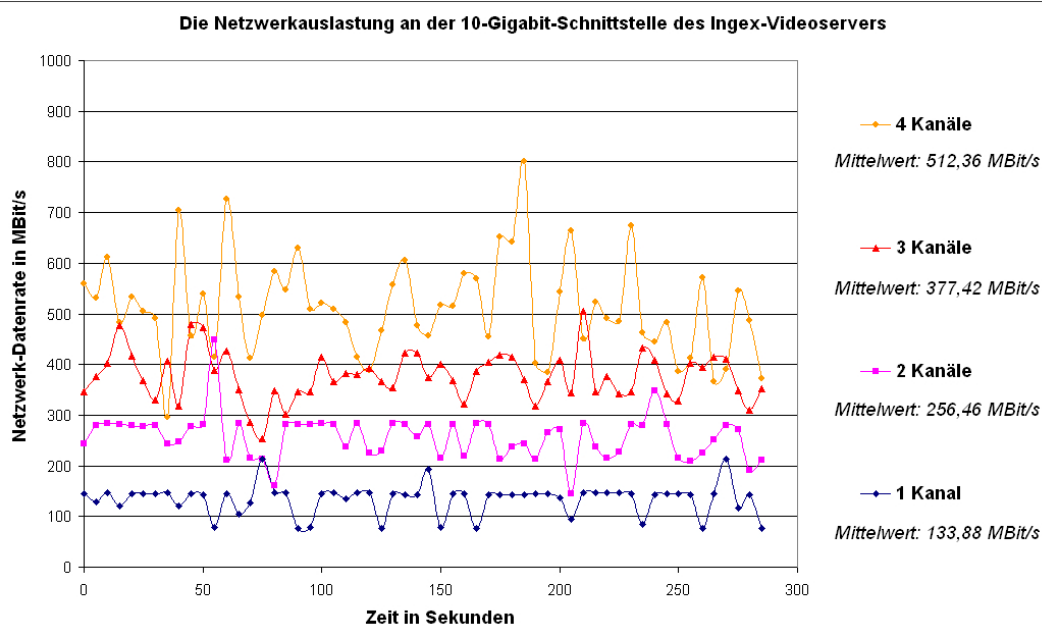


Abbildung 5.6: Protokollierte Werte einer Mehrkanalaufzeichnung über Netzwerk im Codec *DNxHD 120i* (Quelle: eigene Darstellung)

Die Tests zeigen, dass eine 1-Gigabit-Schnittstelle für das Encoding von 4 Kanälen *DNxHD 120* über Netzwerk ausreichend ist. Die maximale Datenrate betrug 800 MBit/s. Damit bleiben 20% Puffer übrig. Die Gründe für die hohen Schwankungen der Datenrate ließen sich im Rahmen des Testaufbaus nicht feststellen. Die Mittelwerte zeigen, dass zudem immer ein nomineller Overhead (sonstiger Netzwerkverkehr u.Ä.) vorhanden ist. Für Kompressionsformate mit höherer Datenrate wie z.B. *DNxHD 185i* halte ich 1 GBit/s für zu knapp bemessen, sobald 4 Kanäle codiert werden müssen.

¹⁷²<http://www.gropp.org/>

5.5.3 Zugriff mehrerer Avid-Clients auf Material per Edit-While-Capture

Ein zweiter Netzwerktest betraf den Zugriff von Schnitt-Systemen. Ich wollte testen, welche Belastungen in einer vernetzten Schnittumgebung auftreten. Dazu analysierte ich die Datenraten der 10-Gigabit-Schnittstelle des EditShare-Speichers, während bis zu 4 Avid-Clients gleichzeitig 2-3 einlaufende HD-Ströme non-linear bearbeiteten. Der Zugriff fand im Edit-While-Capture-Betrieb statt. In jedem Avid wurden die Eingangssignale mit einem Bild-in-Bild-Effekt (*Picture-In-Picture*, siehe auch Abbildung 5.7) in eine Sequenz geschnitten und simultan abgespielt. Über den Effekt konnten bis zu 12 parallele Lesezugriffe über Netzwerk simuliert werden, während 3 Feeds über die selbe Leitung geschrieben wurden.

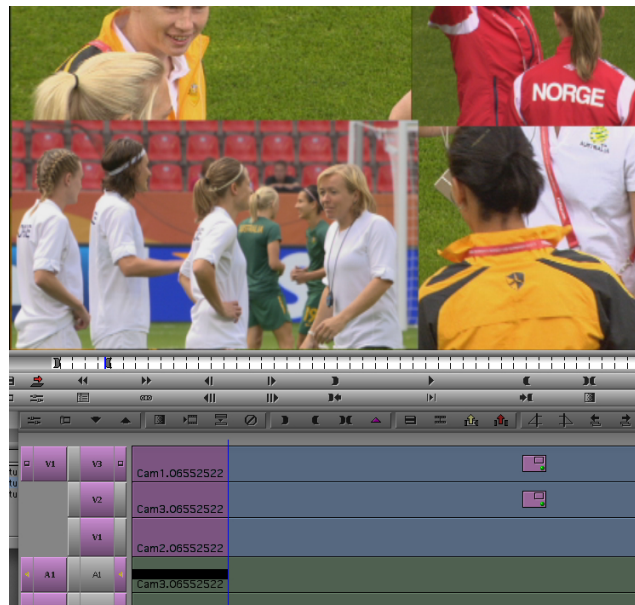


Abbildung 5.7: Die anwachsenden MXF-Dateien wurden zeitgleich im Avid bearbeitet und abgespielt (Quelle: Screenshot)

Die Netzwerkauslastung wurde mit dem grafischen Werkzeug *GKrellm*¹⁷³ gemessen. Die Gesamtlast an der 10-Gigabit-Schnittstelle sollte eine Tendenz aufzeigen, mit welchen Datenraten bei der Echtzeit-Nutzung von 2 bzw. 3 kodierten DNxHD 120i Strömen durch Schnittprogramme zu rechnen ist. Die Werte dienen als Orientierung und stellen keine ausführliche statistische Erhebung dar.

¹⁷³<http://members.dslextreme.com/users/billw/gkrellm/gkrellm.html>

5 Einsatz in der Außenübertragung

Zahl der Avid-Clients	Datenrate für 2 Kanäle (MBit/s)	Datenrate für 3 Kanäle (MBit/s)
0	240-280	320-480
1	440-560	640-880
2	680-800	1040-1280
3	1000-1200	1200-1680
4	1200-1600	1600-2240

Tabelle 5.3: Datenraten an der Netzwerk-Schnittstelle des EditShare-Speichers für 2-3 DNxHD120-Aufzeichnungskanäle und bis zu 4 Avid-Clients (Quelle: eigene Darstellung)

Die Ergebnisse (Tabelle 5.3) zeigen deutlich, dass für Edit-While-Capture mit mehr als einem Avid-Schnittsystem die 10-Gigabit-Schnittstelle unumgänglich ist, wenn multiple Signale aufgezeichnet und bearbeitet werden. Zumindest ist das für die hier genutzte Netzwerkstruktur und Versuchsanordnung festzustellen. Bei 2 Eingangssignalen sind maximal 2 Schnitt-Clients über eine 1-Gigabit-Schnittstelle möglich, bei 3 Eingangssignalen lediglich einer.

6 Ausblick

Ingex soll künftig einer breiteren Masse an Anwendern zugänglich gemacht werden. Über die Plattform *Ingex Solutions*¹⁷⁴ möchte die BBC private Firmen unterstützen, die die Software oder Teile davon einsetzen. Dadurch erhofft sie sich einen gegenseitigen Wissenstransfer und Weiterentwicklung.¹⁷⁵

Zunächst muss die Software jedoch aus ihrem relativ rohen Entwicklungsstand heraus und leichter zu implementieren sein. Vielen Firmen fehlt es an Zeit und Personal, um auf Open-Source-Lösungen zurück zu greifen und diese mühsam anzupassen. Dieser Schritt muss vereinfacht werden. Möglicherweise könnten spezialisierte Dienstleister zukünftig Ingex-Systeme auf den jeweiligen Anwender zuschneiden.

Open-Source-Software ist abhängig von stetiger Weiterentwicklung. Zwar steht mit der BBC eine große europäische Fernsehanstalt hinter Ingex - das Entwicklungsteam ist jedoch klein. Damit geht ein Anwender das Risiko ein, dass an der Software irgendwann nicht weiter gearbeitet wird. Zudem: Service ist möglich und wird von den BBC-Mitarbeitern häufig schnell geleistet, lässt sich aber bei freier Software nicht einfordern.

Der offene Quellcode ermöglicht allerdings, dass selbstständig Veränderungen vorgenommen werden. Erweiterungen wie neue Codecs können theoretisch problemlos implementiert werden. Zum Beispiel könnte der hochwertige *AVC-Intra* Codec von Panasonic mit einem entsprechendem Software Development Kit (SDK) Einsatz finden. Firmen wie Mainconcept bieten solche SDKs an.¹⁷⁶

Die BBC ist stetig darum bemüht, aktuelle Aufzeichnungs-Codecs zu implementieren. Für den DVCPRO-HD Codec engagierten sie einen Entwickler. Momentan wird der XDCAM-422 HD Codec zudem auch für den MXF-

¹⁷⁴Vgl. <http://ingex.tv/>

¹⁷⁵Vgl. WHP186 S. 15

¹⁷⁶Vgl. Mainconcept: H.264/AVC Encoder Broadcast, unter:
<http://www.mainconcept.com/de/produkte/sdks/video/h264avc.html> (abgerufen am 08.07.2011)

6 Ausblick

OP1a-Container in die Software eingebunden. Der AVC-Intra-Codec könnte demnächst mittels einer neuen Bibliothek in Ingex integriert werden.¹⁷⁷

¹⁷⁷Angaben von Philip de Nier auf Anfrage, BBC-Entwickler, 10.07.2011

7 Fazit

Der Aufbau und Einsatz des IngeX-Prototyps im Umfeld von Fernsehproduktionen zeigt: Open-Source-Software und Standard-IT bieten einsatzfähige und vor allem günstige Alternativen zu proprietären Aufzeichnungssystemen. Ein IngeX-Video-Server kann zur mehrkanaligen Aufzeichnung für non-lineare Schnittsysteme genutzt werden. Er kann ebenso eingesetzt werden, um ein bestehendes System durch zusätzliche Kanäle zu ergänzen. Erfordernisse wie taugliche Broadcast-Codecs und effiziente Workflows werden erfüllt, auch für HDTV-Produktionen.

Die Voraussetzung für den Einsatz von IngeX sind weitreichende Computer-Kenntnisse des Administrators eines solchen Systems. Das Einrichten und Konfigurieren ist noch mit hohem Aufwand verbunden. Die Portierung des Systems auf Debian brachte Vorteile für die Administration des Video-servers, erforderte aber komplexe Anpassungen. Die IngeX-Software ist in ständiger Entwicklung, daher müssen kleinere Fehler selbstständig behoben werden.

Gleichzeitig bieten sich durch den offenen und modularen Aufbau zahlreiche Erweiterungsmöglichkeiten. Dadurch war es mir möglich, einen Edit-While-Capture-Workflow für Avid-Schnittsysteme einzurichten. Die offenen, standardisierten Formate MXF und AAF sind zudem bestens geeignet, um Media- und Metadaten in Echtzeit aufzunehmen und in eine non-lineare Schnittumgebung zu überführen.

Es zeigte sich, dass mehrfache Enkodiervorgänge von Quellsignalen parallel möglich sind. Damit lässt sich Material gleichzeitig in verschiedenen Qualitätsstufen wie Online und Offline, sowie zur Sichtung durch einen Redakteur aufzeichnen. Der modulare Aufbau ermöglicht eine einfache Erweiterung der Kapazitäten. Das Potential von IngeX ist somit enorm. Zahlreiche Aufzeichnungskanäle lassen sich für vergleichsweise geringe Kosten einrichten.

IngeX ist ein System, dass sich sehr gut für Mehrkamera-Aufzeichnungen bei Live-Events einsetzen lässt. Von Konzertaufzeichnungen für die DVD-

7 Fazit

Produktion im hochwertigen DNxHD 185 Codec bis zur Sport-Produktion mit hohem zeitlichen Produktionsdruck und Edit-While-Capture bietet sich eine Reihe an Möglichkeiten. Zahlreiche Metadaten (Cue-Punkte, Kommentare, Timecodes), Multicam-Funktion und der Vorschnitt durch den *Directors Cut* (siehe Punkt 3.4.8) vereinfachen und verschnellern die Produktionsabläufe.

Dank der Anpassungen ist Edit-While-Capture für Avid-Schnittsysteme möglich. Dabei muss nicht wie üblich auf eine Avid Interplay Umgebung oder Avid Storages (ISIS/Unity/Medianet) zurückgegriffen werden. Standalone-Schnittplätze können so kostensparend EWC nutzen. Wenn Projekte von vielen Avid-Clients geteilt werden müssen, kann auf separate Lösungen zurückgegriffen werden.

Ein Videoserver mit Standard-IT und Open-Source-Software bringt im Ergebnis deutliche Kosteneinsparungen. In Verbund mit der Möglichkeit zum Edit-While-Capture lässt sich auf Produktionen außerdem beträchtlich Zeit sparen. Damit ist ein solches System insbesondere für kleinere mittelständische Firmen überlegenswert.

8 Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen angefertigt habe.

Die Arbeit wurde noch nicht veröffentlicht oder einer anderen Prüfungsbehörde vorgelegt.

Leipzig, 22.07.2011

Richard Kästner

9 Danksagung

Für die freundliche Unterstützung und die konstruktiven Hinweise möchte ich mich bei meinen Betreuern Prof. Dr.-Ing. Rainer Zschockelt von der Hochschule Mittweida und Dipl.-Ing. Jens Hoppe vom Mitteldeutschen Rundfunk bedanken.

Ein besonderer Dank geht an Andreas Jasper, Schnitt-Administrator der Media Mobil GmbH, der das Projekt vorangetrieben und mir bei der Umsetzung des Themas geholfen hat. Ebenfalls Dank sagen möchte ich dem BBC-Mitarbeiter Philip de Nier, der mir half, das Ingex-System nach meinen Vorstellungen anzupassen.

Literaturverzeichnis

Monografien

- [AMBERG] Amberg, Eric: Linux-Server mit Debian GNU/Linux - Das umfassende Praxis-Handbuch, Redline GmbH Heidelberg, 1. Auflage, 2007
- [APPLE] Apple: QuickTime File Format, Apple Computer Inc., California (USA), 2001
- [AUSTERBERRY] Austerberry, David: Digital Asset Management, Professional Video and Television File-based Libraries, Focal Press, Second Edition, 2006
- [AVID] Avid: MXF Unwrapped, Avid White Paper MUWP0606, Avid Post Production, 2006
- [AVID2] Avid: Unity MediaManager, User's Guide 1.0, Avid Technology Inc., 1999
- [AVID3] Avid: Interplay - Best Practices, Avid Technology Inc., 2007
- [BBC] Tudor, P.N.; Cunningham, S.H.: BBC White Paper 133 „Improving workflow in practice for low cost programme-making using MXF & AAF file formats“, British Broadcasting Corporation, 2006
- [BBC2] Fletcher, J.; Kirby, D.G., Cunningham, S.: BBC White Paper 141 „Tapeless and paperless: automating workflow in tv studio production“, British Broadcasting Corporation, 2006
- [BBC3] Cunningham, Stuart; de Nier, Philip: BBC White Paper 155 „File-based Production: Making It Work In Practice“, British Broadcasting Corporation, 2007
- [BBC4] Tudor, P.N.; McKinnell, J.S.; Pinks, N.P.; Casey, S.: BBC White Paper 186 „Developments in Automated Tapeless Production for Multi-Camera Programmes“, British Broadcasting Corporation, 2010

Literaturverzeichnis

- [BBC5] BBC: Ingeg Installation Guide „Building the Ingeg tapeless recording software“, British Broadcasting Corporation, 2010
- [CHANTELAU/BROTHUHN] Prof. Dr. Chantelau, Klaus; Brothuhn, René: Technologien zur Entwicklung multimedialer Client-Server-Systeme, Springer-Verlag Berlin Heidelberg, 2010
- [CROOKS] Crooks, Roger: Grass Valley Application Note „Interfacing K2 to Avid Editing Systems - Example Workflows with K2 and Avid Interplay and Avid Standalone Editing Systems“, Grass Valley, 2009
- [DEVLIN] Devlin, Bruce: MXF AS02 Introduction, Work In Progress amongst AMWA members, AMWA, 2009
- [EBNER] Ebner, Andreas: Welche Bedeutung haben Metadaten? Anwendung des BMF-Datenmodells, Institut für Rundfunktechnik (IRT), 2008
- [FERREIRA] Ferreira, Pedro: MXF - a progress report, EBU Technical Review, European Broadcasting Union, Geneva, 2010
- [GERLOFF] Gerloff, Karsten: Low-cost high tech - BBC tries out Open Source-based tapeless recording, Fallstudie, UNU-MERIT, Open Source Observatory and Repository (OSOR), 2008
- [GLÄSS/FLÜGEL/RÖHR/R.] Gläß, M.; Flügel, M.; Röhr, N.; Rygol, R.: Material eXchange Format vs. Vergleichbare Formate, Studienarbeit, GRIN Verlag, 1. Auflage, 2008
- [LIEBL/WALLASZKOVITS] Liebl, Christian; Wallaszkovits, Nadja: UMID - Unique Material Identifier, Briefing Paper, Austrian Academy of Sciences, Digital Preservation Europe, 2009
- [MÜCHER] Mücher, Michael: BET Broadcast Fachwörterbuch, Verlag BET Michael Mücher, 16. Auflage, 2008
- [NOWAK/RÖDER] Nowak, Arne; Röder, Jan: „Möglichkeiten des MXF-Formates bei der parallelen Produktion für verschiedene Produktionskanäle mit Virtual Set Systemen“, TU Ilmenau, Institut für Medientechnik, FG Audiovisuelle Technik, in: Elektronische Medien, Fachbericht 188, VDE Verlage, 2005
- [NUFER] Nufer, Christoph: EBU webinar on MXF, Productions Systems Television, Institut für Rundfunktechnik, 2011

Literaturverzeichnis

- [OEHRMANN/JÄSCHKE/D.] Oermann, Andrea; Jäschke, Gerald; Dittmann, Jana: Vertrauenswürdige und abgesicherte Langzeitarchivierung multimedialer Inhalte, Otto-von-Guericke Universität Magdeburg, Network of Expertise in Long Term Storage of Digital Resources, 2009
- [SANTOS] Santos, Ernesto: Operational Patterns ... the MXF Flavors, MOG Solutions, Portugal, 2007
- [SCHMIDT] Heyna, Arne; Briede, Marc; Schmidt, Ulrich: Datenformate im Medienbereich, Fachbuchverlag Leipzig, 2003
- [SCHROEDER/GIBSON] Schroeder, Bianca; Gibson, Garth A.: Disk failures in real world: What does an MTTF of 1,000,000 hours mean to you?, 5th USENIX Conference on File and Storage Technologies (FAST 07), 2007
- [SMPTE] SMPTE EG41: Material Exchange Format (MXF), Proposed SMPTE Engineering Guideline for Television, The Society of Motion Picture and Television Engineers (SMPTE), New York, 2003
- [WELLS] Wells, Nick: The MXF Book, Boston: Focal Press, 2006

Zeitschriften

- [HEDTKE/SCHNÖLL] Hedtke, Rolf; Schnöll Matthias: „Verteilte Plattform für Produktion und Archiv“, in: FKT 1-2/2010, S. 21-28, Fachverlag Schiele & Schön GmbH, 2010
- [SINGLETON] Singleton, Alex: „Linux at the BBC“, in: Linux Format LXF105 May/08, 2008
- [WALSH/GEPPERT] Walsh, Bernie; Dr. Geppert, Thomas: „Storage Management - digitale Inseln verbinden“, in: FKT 12/2006, S. 750-754, Fachverlag Schiele & Schön GmbH, 2006

Internetseiten

- [AAF] Projektseite des AAF Toolkits / SDK, unter: <http://aaf.sourceforge.net/>
- [ALPERMANN+VELTE] Alpermann+Velte: Ancillary Time Code (ATC) according to SMPTE 12M-2, unter: http://www.alpermann-velte.com/faq_e/e_atc.html (abgerufen am 20.06.2011)

Literaturverzeichnis

- [DIEDRICH] Diedrich, Oliver: Trendstudie Open Source - Eingesetzte Produkte, Heise Zeitschriften Verlag, unter: <http://www.heise.de/open/artikel/Trendstudie-Open-Source-221696.html>, 2009 (abgerufen am 21. Juni 2011)
- [LINDNER] Lindner, Mirko: libav soll FFmpeg ersetzen, unter: <http://www.pro-linux.de/news/1/16813/libav-soll-ffmpeg-ersetzen.html> (abgerufen am 19.06.2011)
- [INGEX] Ingex Projekt Forum: Entwickler-Einträge, unter: <http://sourceforge.net/projects/ingex/forums/>, Links und Abrufdatum in Fußnoten
- [INGEX2] Ingex: Hardware Requirements, unter <http://ingex.sourceforge.net/studio/> (abgerufen am 23.06.2011)
- [IRT] Institut für Rundfunktechnik: MXF Interoperability Plug-Fest, unter: <http://mxf.irt.de/activities/2010-11-MxfPlugFest.php> (abgerufen am 28.06.2011)
- [IRT2] Institut für Rundfunktechnik: Standardisierter Metadaten austausch mit BMF 2.0, unter <http://www.irt.de/de/themengebiete/produktion/bmf.html> (abgerufen am 18.06.2011)
- [MAINCONCEPT] Mainconcept: H.264/AVC Encoder Broadcast, AVC-Intra Class 50 and 100, unter: <http://www.mainconcept.com/de/produkte/sdks/video/h264avc.html> (abgerufen am 08.07.2011)
- [SONY] Sony History: The Beginning of the Digital Age, unter: <http://www.sony.net/SonyInfo/CorporateInfo/History/SonyHistory/2-04.html> (abgerufen am 27.06.2011)

Empfehlungen

- [SMPTE RP 210.4-2002] SMPTE, Metadata Dictionary Registry of Metadata Element Descriptions
- [SMPTE RP 224] SMPTE, SMPTE Labels Registry

Normen

- [SMPTE 330M-2004] SMPTE, Unique Material Identifier (UMID)

Literaturverzeichnis

[SMPTE 336M-2004]	SMPTE, Data Coding Protocol using Key-Length-Value
[SMPTE 377M]	SMPTE, Material Exchange Format (MXF) - File Format Specification
[SMPTE 378M-2004]	SMPTE, MXF Operational Pattern 1a
[SMPTE 380M-2004]	SMPTE, MXF Descriptive Metadata Scheme 1
[SMPTE 390M/ST 0390-2011]	SMPTE, Material Exchange Format (MXF) - Specialized Operational Pattern „Atom“

Abbildungsverzeichnis

2.1	Der hierarchische Aufbau eines Speichersystems (Quelle: eigene Darstellung)	9
2.2	Basisstruktur einer beispielhaften MXF-Datei (Quelle: Wells S. 24)	17
2.3	Das Material Package (oben) mit den abzuspielenden Tracks und das File Package (unten) mit der Originalessenz im Vergleich (Quelle: Wells S. 27)	18
2.4	Frame-Wrapping (oben) und Clip-Wrapping (unten) mit ihrer unterschiedlichen KLV-Struktur (Quelle: Nufer S. 10)	19
2.5	Die Generalized Operational Patterns mit ihrer unterschiedlichen Paket- und Abspielkomplexität im Überblick (Quelle: SMPTE Engineering Guideline S. 38)	21
2.6	Die Pfadstruktur des AS02 (Quelle: Devlin S. 3)	27
2.7	Eltern- und Kindatome im QuickTime-Container (Vgl. Apple: QuickTime File Format)	29
2.8	Im Idealfall werden alle Bereiche der Fernsehproduktion vom BMF bedient (Vgl. Ebner S. 4)	32
3.1	Ingex im System eines vernetzten Studios	37
3.2	Der Ring Buffer speichert die eingehenden Daten zwischen (Quelle: Wikipedia)	41
3.3	Ausgabe des Capture-Prozesses von Ingex (Quelle: Screenshot auf Ingex-PC)	42
3.4	Schematische Darstellung einer Aufzeichnung von 4 SDI-Quellen mit 2 Recordern (Quelle: eigene Darstellung)	45
3.5	Das Ausgabefenster eines Recorders nach einer abgeschlossenen Aufnahme (Quelle: Screenshot auf Ingex-PC)	45
3.6	Ingex-GUI mit Bedienoberfläche, Quadsplit, Capture- und Recorderprozess (Quelle: Screenshot auf Ingex-PC)	46

Abbildungsverzeichnis

3.7	Ausschnitte der Ingexgui während (a) und nach (b) der Aufnahme (Quelle: Screenshots auf Ingex-PC)	47
3.8	Die Ingex-Weboberfläche mit dem Materialbrowser (Quelle: Screenshot auf Ingex-PC)	49
3.9	Recorder-Einstellung in der Weboberfläche (Quelle: Screenshot auf Ingex-PC)	50
3.10	Die Systemkapazitäten sind grafisch dargestellt (Quelle: Screenshot auf Ingex-PC)	50
3.11	Directors Cut nutzt Bildtexturmessung (Frame Matching), um Signalquellen abzugleichen (Quelle: BBC White Paper 186) .	51
3.12	Bis zu 4 Videosignale und 16 Audiospuren lassen sich im Quadsplit darstellen (Quelle: Screenshot auf Ingex-PC) . . .	52
4.1	Beispiel eines Netzwerkes aus Recorder (Ingex), Speicher (NAS) und Clients (Avid) (Quelle: BBC White Paper 155) . . .	60
5.1	InfoDumper-Auszug aus einem AAF - Die EWC-Attribute liegen als Metadaten vor (Quelle: eigene Darstellung)	71
5.2	Avid stellt anwachsende MXF-Dateien mit In-progress Clips dar (Quelle: Avid)	73
5.3	Capture in Progress im Avid - Bereits verfügbares Material wird angezeigt (links), während noch der noch nicht aufgezeichnete Teil (rechts) „in Progress“ ist (Quelle: Screenshot auf Avid-PC)	74
5.4	Schematische Darstellung der Versuchsanordnung für 2 HD-Kanäle (Quelle: eigene Darstellung)	78
5.5	Die Recorderprozesse während des Ingests mit bis zu 4 HD-Kanälen (Quelle: eigene Darstellung)	79
5.6	Protokollierte Werte einer Mehrkanalaufzeichnung über Netzwerk im Codec DNxHD 120i (Quelle: eigene Darstellung) . .	81
5.7	Die anwachsenden MXF-Dateien wurden zeitgleich im Avid bearbeitet und abgespielt (Quelle: Screenshot)	82
.1	Der mit MXFDump ausgelesene Header einer MXF-Datei (Quelle: eigene Darstellung)	99
.2	Der mit MXFDump ausgelesene Footer einer MXF-Datei (Quelle: eigene Darstellung)	100

Tabellenverzeichnis

2.1	Eigenschaften der einzelnen Speichersysteme (Quelle: Aus- terberry S. 265)	11
3.1	Aufzeichnungs- und Schnittsysteme im Vergleich (Quelle: ei- gene Darstellung)	54
4.1	Einige Komponenten des Ingex-Prototyps im Überblick	62
5.1	MXF OP-Atom während der Aufzeichnung mit EVS und In- gex, verglichen mit einem abgeschlossenen File (Quelle: ei- gene Darstellung)	69
5.2	Die durchschnittlichen CPU-Lasten für 1-4 Kanäle im Ver- gleich mit der jeweiligen Spitzenlast (Quelle: eigene Darstel- lung)	80
5.3	Datenraten an der Netzwerk-Schnittstelle des EditShare-Speichers für 2-3 DNxHD120-Aufzeichnungskanäle und bis zu 4 Avid- Clients (Quelle: eigene Darstellung)	83
.1	Verfügbare Codecs von Ingex im Überblick (Vgl. Kapitel 3.5) .	101

Anhang

1	Dump an MXF file using MXFLib	
2	- using dictionary "/var/ingexpakete/MXFlib/mxfLib-1.0.1/dict.xml"	
3		
4	Partition at 0x00000000 is for BodySID 0x0000	
5	ClosedCompleteHeader	
6	MajorVersion = 1	
7	MinorVersion = 2	
8	KAGSize = 256	
9	ThisPartition = 0	
10	PreviousPartition = 0	
11	FooterPartition = 423231520	
12	HeaderByteCount = 261920	
13	IndexByteCount = 0	
14	IndexSID = 0	
15	BodyOffset = 0	
16	BodySID = 0	
17	OperationalPattern = MXF Specialized OP Atom	
18	EssenceContainers	
19	EssenceContainer = MXF-GC DV-DIF Mappings	
20		
21	Header Metadata:	
22	Unknown {b3b398a5-1c90-11d4-8053-080036210804} = 3c 0a 00 10 73 ba 2e c7 3f 04 4d 91 af 8c 58 8c 1a 78 3b 63 00 03 00 18 00 00 00 00 00 00 00 00 00	
23	Unknown {060e2b34-0253-0101-0d010101-02250000} = 3c 0a 00 10 d6 c8 4a 8f f5 e1 40 16 92 87 98 87 15 e6 37 c4 00 03 03 68 00 00 36 00 00 00 10 08 :	
24	Unknown {060e2b34-0253-0101-0d010101-02010000} = 3c 0a 00 10 08 31 ef 54 52 fe 41 6a 90 ce 46 57 6a b8 19 85 00 05 00 10 06 0e 2b 34 02 53 01 01 0d :	
25	Unknown {060e2b34-0253-0101-0d010101-02020000} = 3c 0a 00 10 9e 81 9e 35 b0 f3 47 e b5 d1 29 86 3e dd b5 e1 00 05 00 10 06 0e 2b 34 01 01 02 05 :	
26	Unknown {060e2b34-0253-0101-0d010101-02020000} = 3c 0a 00 10 99 aa 75 a0 ab 92 4d 03 b4 9c 85 22 19 e2 50 ab 00 05 00 10 06 0e 2b 34 01 01 02 06 :	
27	Unknown {060e2b34-0253-0101-0d010101-02010000} = 3c 0a 00 10 6b 0e c6 80 f4 43 48 a9 ad 90 ce 10 3a 36 8e 70 00 05 00 10 06 0e 2b 34 02 53 01 0d :	
28	Unknown {060e2b34-0253-0101-0d010101-02020000} = 3c 0a 00 10 09 51 75 64 b7 4a 58 9f c3 38 79 3b 6b 44 c3 00 05 00 10 06 0e 2b 34 01 01 02 04 :	
29	Unknown {060e2b34-0253-0101-0d010101-02020000} = 3c 0a 00 10 77 d1 7a 03 0c c6 49 74 95 6c b7 91 fc 1d 77 38 00 05 00 10 06 0e 2b 34 01 01 02 07 :	
30	Unknown {060e2b34-0253-0101-0d010101-02020000} = 3c 0a 00 10 ab e1 0b 3d 5a c4 48 58 a0 1b a3 e6 4a c4 1f c9 00 05 00 10 06 0e 2b 34 01 01 07 03 :	
31	Unknown {060e2b34-0253-0101-0d010101-02020000} = 3c 0a 00 10 e0 d3 ec b1 46 ca 45 0c 8a 35 f1 84 19 39 ba a3 00 05 00 10 06 0e 2b 34 01 01 02 03 :	
32	Unknown {060e2b34-0253-0101-0d010101-02020000} = 3c 0a 00 10 be 4c 28 06 8c 19 48 61 a9 5e f9 46 ec 80 45 4f 00 05 00 10 06 0e 2b 34 01 01 07 03 :	
33	Unknown {060e2b34-0253-0101-0d010101-02010000} = 3c 0a 00 10 e7 ca ca 93 40 0e 47 2b 95 5f ad 44 48 34 53 a1 00 05 00 10 06 0e 2b 34 02 53 01 0d :	
34	Unknown {060e2b34-0253-0101-0d010101-02010000} = 3c 0a 00 10 4e 58 3c 29 7b 19 4c c2 a9 7f 5a 07 c5 7a 62 a7 00 05 00 10 06 0e 2b 34 02 53 01 0d :	
35	Unknown {060e2b34-0253-0101-0d010101-02020000} = 3c 0a 00 10 a5 0b b4 94 1f c1 48 d7 bf 54 a3 6d 15 0e bc b8 00 05 00 10 06 0e 2b 34 01 01 02 07 :	
36	Unknown {060e2b34-0253-0101-0d010101-02020000} = 3c 0a 00 10 35 f6 04 2f bd a2 43 95 96 ae 55 4d 98 88 08 d4 00 05 00 10 06 0e 2b 34 01 01 02 05 :	
37	Unknown {060e2b34-0253-0101-0d010101-02010000} = 3c 0a 00 10 f1 b9 7e 90 38 e8 4f 77 b4 a9 25 e7 eb 1b 6e 3a 00 05 00 10 06 0e 2b 34 02 53 01 0d :	

Abbildung .1: Der mit MXFDump ausgelesene Header einer MXF-Datei (Quelle: eigene Darstellung)

1015		InstanceUID = {10f2f1ee-c24d-40c5-9382-a3da33c0a1d0}
1016		DateDefinition = SMPTE 12M Timecode Track
1017		Duration = 10800000
1018		RoundedTimecodeBase = 25
1019		DropFrame = 0
1020		StartTimecode = 0
1021		Descriptor = {a2a1e8ef-5ad0-47a7-9023-3015aa6fc586}
1022		Unknown ff.da {60958183-47b1-11d4-a01c-0004ac969f50} = 00 00 00 01 00 00 10 2e f3 34 ce 51 21 4a dd bb 92 cf f9 45 b5 22 dd
1023		Unknown [060e2b34.0253.0101.0d010101.01012e00] = 3c 0a 00 10 a2 a1 e8 ef 5a d0 47 a7 90 23 30 15 aa 6f c5 86 ff e2 00 04 00 00 00
1024		Unknown [060e2b34.0253.0101.0d010101.01013f00] = 3c 0a 00 10 2e f3 34 ce 51 21 4a dd bb 92 cf f9 45 b5 22 dd 50 01 00 08 00 5f 00 50 00 4a 00 00 50
1025		Unknown {f10296f0-56e0-4d2a-9613-b38a87348746} = 00 00 00 00 02 53 19 73 b4 2e c7 3f 04 d4 91 af 8c 58 dc 1a 78 3b 63 00 00 00 00 14 00
1026		
1027		No index table in this partition
1028		
1029		Partition at 0x00040020 is for BodySID 0x0001
1030		
1031		Partition at 0x193a0020 is for BodySID 0x0000
1032		CompleteFooter
1033		MajorVersion = 1
1034		MinorVersion = 2
1035		KAGSize = 256
1036		ThisPartition = 423231520
1037		PreviousPartition = 262176
1038		FooterPartition = 423231520
1039		HeaderByteCount = 0
1040		IndexByteCount = 313
1041		IndexSID = 2
1042		BodyOffset = 0
1043		BodySID = 0
1044		OperationalPattern = MXF Specialized OP Atom
1045		EssenceContainers
1046		EssenceContainer = MXF-GC DV-DIF Mappings
1047		No header metadata in this partition
1048		
1049		Index Table Segment (first edit unit = 0, duration = 1468) :
1050		Indexing BodySID 0x0001 from IndexSID 0x0002
1051		

Abbildung .2: Der mit MXFDump ausgelesene Footer einer MXF-Datei (Quelle: eigene Darstellung)

Tabellenverzeichnis

Videokompressionsformat	Containerformat	Verwendeter Puffer in Inge ^{x 1}
Unkomprimiert	Raw	UYVY
Unkomprimiert	MXF OP-Atom	UYVY
DV25	Raw	DV25
DV25	MXF OP-Atom	DV25
DV25	QuickTime	DV25
DV50	Raw	DV50
DV50	MXF OP-Atom	DV50
DV50	QuickTime	DV50
DVCPRO-HD	Raw	YUV422 (HD)
DVCPRO-HD	MXF OP-Atom	YUV422 (HD)
DVCPRO-HD	QuickTime	YUV422 (HD)
MJPEG 2:1	MXF OP-Atom	YUV422
MJPEG 3:1	MXF OP-Atom	YUV422
MJPEG 10:1	MXF OP-Atom	YUV422
MJPEG 15:1s	MXF OP-Atom	YUV422
MJPEG 20:1	MXF OP-Atom	YUV422
IMX30	MXF OP-Atom	YUV422
IMX40	MXF OP-Atom	YUV422
IMX50	MXF OP-Atom	YUV422
IMX30	MXF OP-1a	YUV422
IMX40	MXF OP-1a	YUV422
IMX50	MXF OP-1a	YUV422
VC3-36p ²	MXF OP-Atom	YUV422 (HD)
VC3-120i	MXF OP-Atom	YUV422 (HD)
VC3-185i	MXF OP-Atom	YUV422 (HD)
VC3-120p	MXF OP-Atom	YUV422 (HD)
VC3-185p	MXF OP-Atom	YUV422 (HD)
MPEG2 422 Long GOP 50 Mbit/s ³	Raw	YUV422 (HD)
MPEG2 422 Long GOP 50 Mbit/s	QuickTime	YUV422 (HD)
MPEG2 for DVD	MPG	MPEG
MPEG4 Quicktime	QuickTime	MPEG
MP3 Audio only	MP3	alle

[1] Inge^x-eigene Bezeichnungen

[2] VC3 ist der SMPTE-Standard für den DNxHD-Codec von Avid

[3] allgemeine Beschreibung für den XDCAM-HD 422 Codec

Tabelle .1: Verfügbare Codecs von Inge^x im Überblick (Vgl. Kapitel 3.5)